TEMPORAL PROCESSING OF NEWS: ANNOTATION OF TEMPORAL EXPRESSIONS, VERBAL EVENTS AND TEMPORAL RELATIONS

Georgiana Marşic

A thesis submitted in partial fulfilment of the
requirements of the University of Wolverhampton
for the degree of Doctor of Philosophy

2011

This work or any part thereof has not previously been presented in any form to the University or to any other body whether for the purposes of assessment, publication or for any other purpose (unless otherwise indicated). Save for any express acknowledgments, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

Signature .....
Date .....

To my loved ones

Abstract

The ability to capture the temporal dimension of a natural language text is essential to many natural language processing applications, such as Question Answering, Automatic Summarisation, and Information Retrieval. Temporal processing is a field of Computational Linguistics which aims to access this dimension and derive a precise temporal representation of a natural language text by extracting time expressions, events and temporal relations, and then representing them according to a chosen knowledge framework.

This thesis focuses on the investigation and understanding of the different ways time is expressed in natural language, on the implementation of a temporal processing system in accordance with the results of this investigation, on the evaluation of the system, and on the extensive analysis of the errors and challenges that appear during system development. The ultimate goal of this research is to develop the ability to automatically annotate temporal expressions, verbal events and temporal relations in a natural language text.

Temporal expression annotation involves two stages: **temporal expression identification** concerned with determining the textual extent of a temporal expression, and **temporal expression normalisation** which finds the value that the temporal expression designates and represents it using an annotation standard. The research presented in this thesis approaches these tasks with a knowledge-based methodology that tackles temporal expressions according to their semantic classification. Several knowledge sources and normalisation models are experimented with to allow an analysis of their impact on system performance.

The annotation of events expressed using either finite or non-finite verbs is addressed with a method that overcomes the drawback of existing methods

which associate an event with the class that is most frequently assigned to it in a corpus and are limited in coverage by the small number of events present in the corpus. This limitation is overcome in this research by annotating each WordNet verb with an event class that best characterises that verb.

This thesis also describes an original methodology for **the identification of temporal relations** that hold among events and temporal expressions. The method relies on sentence-level syntactic trees and a propagation of temporal relations between syntactic constituents, by analysing syntactic and lexical properties of the constituents and of the relations between them.

The detailed evaluation and error analysis of the methods proposed for solving different temporal processing tasks form an important part of this research. Various corpora widely used by researchers studying different temporal phenomena are employed in the evaluation, thus enabling comparison with state of the art in the field. The detailed error analysis targeting each temporal processing task helps identify not only problems of the implemented methods, but also reliability problems of the annotated resources, and encourages potential reexaminations of some temporal processing tasks.

Acknowledgements

The completion of my doctoral studies has been the most significant academic challenge I was ever confronted with. It has been a long journey whose course was sometimes deterred by life getting in the way, but to whose successful completion many people have contributed directly or indirectly, and I would like to take this opportunity to thank them.

First of all, I would like to thank my supervisory team, Ruslan Mitkov, John Prager and Constantin Orăsan, for their trust, encouragement, patience and guidance. I am thankful to Ruslan Mitkov, my director of studies, for making this thesis possible by providing the necessary infrastructure and resources to accomplish my research work, and for acting as my supervisor despite his many other academic and professional commitments. I would like to thank John Prager for finding the time to read my thesis and to provide insightful and creative comments. I am extremely indebted to Constantin Orăsan, my supervisor, colleague and friend, for always showing a sincere interest in my work, for his constructive criticism, for the extensive discussions concerning my work, and for all the help he has given me throughout my years in Wolverhampton.

I would like to express my special gratitude and appreciation to my former research advisor Dan Cristea who introduced me to the world of Natural Language Processing. I still think fondly of my time as a postgraduate student that I have spent working with him.

I am privileged for having had Verginica Barbu Mititelu, Iustin Dornescu, Richard Evans, Le An Ha, Laura Hasler, Iustina Ilisei, Irina Temnikova, and Andrea Varga as my colleagues, friends and collaborators during my years spent in Wolverhampton. Their friendship and professional collaboration meant a great

deal to me. My special thanks go to Verginica for her cooperation and help with the annotation of several resources that were essential in my research.

I gratefully acknowledge the members of the Research Group on Natural Language Processing and Information Systems from the University of Alicante, Spain for offering me the opportunity to spend one year collaborating with them. I have a fond recollection of the year I worked in the GPLSI group.

I wish to thank Aidan Byrne, Alison Carminke and Erin Stokes for their help with proof-reading my thesis.

I would like to convey my heartfelt thanks to my friends who probably realised that pursuing a PhD can sometimes be a lonely and isolating experience. In particular, Oana Apostol, Mona Corodeanu, Smaranda Cristea, Gabi Haja, Iustina Ilisei, Ioana Roelake and Anca Zaharia receive my gratitude for their friendship and for being able to ignore my absence from their lives. I am grateful to my close friend Smaranda for her warm friendship, and for always caring and worrying about me. My dear friends Iustina and Javier helped me immensely with their support, encouragement, and all the thoughtful little things they did to take my mind off my PhD-related worries. I thank them for their friendship, for the wonderful times we spent together, and for their hospitality during my recent visits to Wolverhampton. I am very lucky to have such good friends.

Finally, but most importantly, my deepest gratitude goes to my family for all their love and continuous support. My parents, Maricica and Ionel Puşcaşu, have always believed in me and encouraged me to strive for excellence in all that I do. My loving sister, Ana Maria Radu, has been a constant source of moral support, and has always been close to me, despite the distance. My grandmother Veronica Luchian and my uncle Mihai Luchian have always been very supportive, offering me invaluable advice. My wonderful husband Vlad spent many sleepless nights by my side when I was writing up this thesis. He has been an endless source of joy, love, kindness, patience and confidence in me, and thanks to him I did not have to worry about anything else but finalising my PhD. I am heartily thankful to all my loved ones, this thesis would certainly not have existed without them. To them I dedicate this work.

Table of Contents

List of Tables

List of Figures

Abbreviations

ACE – Automatic Content Extraction

BNC – British National Corpus

CRF – Conditional Random Field

DCT – Document Creation Time

DUC – Document Understanding Conference

ETL – Event Target List

FDG – Functional Dependency Grammar

IE – Information Extraction

ILI – Inter-Lingual Index

IR – Information Retrieval

LDC – Linguistic Data Consortium

MBL – Memory-Based Learning

MUC – Message Understanding Conference

NE – Named Entity

NER – Named Entity Recognition

NLP – Natural Language Processing

NP – Noun Phrase

POS – Part Of Speech

PP – Prepositional Phrase

QA – Question Answering

SAS – Set of Ambiguous Subordinators

STAG – Sheffield Temporal Annotation Guidelines

STS – Set of Temporal Subordinators

SVM – Support Vector Machine

TDT – Topic Detection and Tracking

TE – Temporal Expression

TERN – Temporal Expression Recognition and Normalisation

TREC – Text REtrieval Conference

VP – Verb Phrase

Chapter 1

Introduction

The temporal dimension of information is fundamental for reasoning about how the world changes. The world is dynamic in its nature, and time is an important aspect of everything that happens in this world. Things that happen and involve change (**events**), or situations that stay the same for a certain period of time (**states**) are related by their temporal reference. People use the concept of time to place events or states in sequence one after the other, to establish how long an event or a state lasted, and to specify when an event occurred. Time seems to play the role of an universal reference system that is used to anchor, sequence, measure and compare the intervals occupied by events and states.

Recent years have seen unprecedented interest in natural language processing (NLP) applications that can process the wealth of electronic data available, with the need for temporally aware systems becoming increasingly popular. This need is justified by the fact that most of the information available electronically is temporally sensitive, in the sense that something that was true at some point in time could be false at another. Despite its omnipresence, agreeing on how time can be formalised has historically been a difficult task, as well as incorporating it into automatic systems that can access the temporal dimension and extract the temporal meaning of a text, known as **temporal processing systems**.

1.1 About temporal processing

This thesis investigates the area of temporal processing, a topic that has received growing interest in recent years. The ultimate aim of research in this area is the automatic identification of all temporal referring expressions, events, and temporal relations within a text. These tasks are difficult even for humans if they are asked to formalise them in a language understood by computers, despite the fact that they manage temporal information very naturally and efficiently during their everyday life. There are several explanations for this difficulty.

One explanation for the difficulty of identifying temporal information in text is the fact that temporal information can be conveyed via a wide range of different mechanisms including tense, aspect, and lexical semantic knowledge (Mani et al., 2005). These mechanisms need to be correctly identified, interpreted and combined to derive the appropriate temporal information.

Another challenge arises from the fact that temporal information is not always stated explicitly, being often implicit and requiring interpretations or inferences derived from world knowledge. The sentences in [1.1] and [1.2] have similar syntax, but the events they describe are not in the same temporal order.

[1.1] *John fell. Mary pushed him.* [1]

[1.2] *John fell. Mary asked for help.*

The temporal information in these examples is implicit, as the events described are neither anchored to precise points in time, nor specifically ordered with respect to neighbouring events. To derive the correct temporal interpretation for these examples, one must rely on semantic content, knowledge of causation and knowledge of language use. Despite their structure and syntax being so

1. The examples used in this thesis are either created to illustrate a certain phenomenon, or extracted from various sources such as the BNC, annotation guidelines, articles, and so on.

similar, in the first example the event of *falling* is temporally after the event of *pushing*, while in the second example the event of *falling* precedes the event of *asking*.

Computers currently find it extremely difficult to "understand" semantic information of the type required to distinguish between the two examples above, and to infer the correct temporal order in both cases. As a result, the research community has focused mainly on the various mechanisms used by language to convey temporal information explicitly or implicitly, mechanisms that are automatically identifiable using state-of-the-art techniques. Any temporal processing system should possess abilities to identify such mechanisms in text, and exploit them in solving the following tasks: temporal expression identification and normalisation, event annotation, and temporal relation identification. These tasks are illustrated below using an excerpt from a news article.

27/02/1998

OAU to investigate Rwandan genocide

The Organization of African Unity said *Friday* it would investigate the Hutu-organized genocide of more than 500,000 minority Tutsis in Rwanda *nearly four years ago*. Foreign ministers of member-states meeting in the Ethiopian capital agreed to set up a seven-member panel to investigate who shot down Rwandan President Juvenal Habyarimana's plane on *April 6, 1994*.

Temporal Processing Tasks

Task 1: Temporal Expression Identification

e.g. "nearly four years ago"

Task 2: Temporal Expression Normalisation

e.g. "nearly four year ago" => "1994"

Task 3: Event Annotation

e.g. "shot" => "OCCURRENCE"

Task 4: Temporal Relation Identification

e.g. temporal relation between "shot" and "April 6, 1994" is "OVERLAP";
temporal relation between "investigate" and "shot" is "AFTER"

The ability to derive a precise temporal representation of a text by solving these tasks can improve the performance of many practical NLP applications that require access to the temporal dimension of information, as exemplified below.

1.2 Applications of temporal processing in NLP

The development and evaluation of temporal processing systems is not only an important research topic, but also a very practical challenge. Temporal information has become more and more relevant to many NLP applications such as Question Answering (Moldovan et al., 2005; Saurí et al., 2005), Automatic Summarisation (Mani and Shiffman, 2005), Information Retrieval (Alonso et al., 2007), and Information Extraction (Surdeanu et al., 2003).

Question Answering (QA) systems process large text collections to find "a short phrase or sentence that precisely answers a user's question" (Prager et al., 2000). QA systems need temporal processing to answer questions that explicitly request temporal information as their answer (e.g. [1.3]), or questions that manifest an intrinsic time dependency (e.g. [1.4], [1.5] and [1.6]).

[1.3] *When did the French Revolution begin?*

[1.4] *Is Gates currently CEO at Microsoft?*

[1.5] *Who was president of Enron when its share price was highest?*

[1.6] *Did the Enron merger with Dynergy take place?*

While question [1.3] can easily be answered if a candidate paragraph contains an explicit mention of the date the French Revolution started, questions like [1.4], [1.5], and [1.6] cannot be correctly answered unless advanced temporal processing methods are employed to analyse the temporal properties, modalities and ordering of the events involved.

Automatic Summarisation also places increasing demands on the processing of temporal information. Automatic Summarisation systems "take one or several texts and extract the ≪most important≫ information [...] from them" (Orăsan, 2006). Multi-document summarisation of news articles which overlap in their description of events would benefit from knowing the relative order of events. This temporal information is essential for assembling a chronologically coherent narrative from the events mentioned in diverse information sources. Automatic Summarisation has many practical applications that include generating biographies, assisting journalists in preparing background information on breaking news, condensing clinical records and deriving the typical evolution of a disease, and so on.

Information Retrieval (IR) is the field of study concerned with "finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)" (Manning et al., 2008). With the rapid increase in digital information, the concept of time as a dimension through which information can be organised and explored becomes extremely relevant for IR. Access to temporal information could benefit IR tasks such as clustering of search results according to various time attributes (e.g. Google News Timeline [2]), or time-based browsing

2. Available online at: http://news.google.com/

and exploration of search results using timelines (e.g. Inxight's TimeWall [3]).

Information Extraction (IE) is "the name given to any process which selectively structures and combines data which is found, explicitly stated or implied, in one or more texts" (Cowie and Wilks, 2000). The IE focus is typically on extracting attributes of entities (e.g. a person's professional position), or relations between entities (e.g. the *employee_of* relation). In many cases the extracted attributes and relations are valid only within certain temporal boundaries, as entities and their properties change over time, therefore it is important to capture these temporal restrictions to improve the IE process.

Many applications would benefit from obtaining a precise temporal representation of a text, and with all the digital data available it is impossible to add temporal mark-up by hand, therefore the need for reliable temporal processing systems that can automatically perform temporal annotation has never been greater.

1.3 Original contributions of this thesis

This thesis presents a systematic investigation of how temporal information can be identified in natural language texts. The research in this thesis proposes a framework for the identification and resolution of temporal information, illustrating how the tasks of identifying **temporal expressions (TEs)**, **verbal events** and the **temporal relations** holding among them can be automatically addressed.

This study contributes to advances in automatic temporal processing of text in three areas:

3. More information available at: http://www.inxightfedsys.com/products/sdks/tw/default.asp

1. novel methodology;

2. comparative evaluation facilitated by a modular approach;

3. resources for temporal processing.

To achieve this, an extensive review of the existing research in temporal processing is carried out. This review focuses on both linguistic and computational linguistic aspects of the field. Afterwards, a novel methodology for the identification and annotation of temporal information in text is developed and evaluated following extensive corpus-based and corpus-driven investigations. The main contributions of this research are presented below.

The **first main contribution** of this work is the development of a novel methodology in automatic temporal processing which can identify and annotate different types of temporal information in text, such as temporal expressions, verbal events and temporal relations. In the case of temporal expressions, the main contribution consists in tackling temporal expressions according to their semantic classification. The exhaustive classification of temporal expressions guiding this work is also unique in the specialised literature. Another contribution is the methodology for the automatic disambiguation of the temporal adverb *then*. In the case of verbal events, given the drawbacks of existing methods for event classification (e.g. their narrow coverage), a different methodology is proposed for their classification. An event is typically assigned by other researchers the most frequent class associated with it in TimeBank (Pustejovsky et al., 2006), the reference corpus annotated with temporal information. If the event is not mentioned in TimeBank, then no class can be predicted. This is where this research brings a novel contribution, as the developed methodology provides much

better coverage than existing methods. The present work also brings original contributions to the identification of temporal relations by introducing a new methodology that closely follows human behaviour when deciding the temporal relation between two entities.

The **second main contribution** of this research is that it performs a comparative, qualitative and quantitative evaluation of modular temporal processing systems. The purpose of this comparative evaluation is to uncover the influence of different modules on the entire annotation process. In order to assess this, an integrated system that allows one to easily switch modules on and off is used.

The **third main contribution** of this study is the development of novel resources, including:

- a corpus illustrating different usages of *then* for training and testing the methods employed for its disambiguation;

- a resource that links each verb present in WordNet (Fellbaum, 1998) to the event class that best characterises that verb;

- a corpus capturing the behaviour of ambiguous subordinators that are able to introduce temporal clauses, where each subordinator receives an annotation that sets apart a temporal usage from a non-temporal one.

The above contributions are achieved by setting a number of goals that are presented in the following section.

1.4 Research goals

The research presented in this thesis targets the accomplishment of the following goals:

Goal 1 is to provide a comprehensive review of how temporal information is conveyed in natural language from a theoretical perspective. This review is necessary in order to guide the development of automatic systems that target the identification of different types of temporal information.

Goal 2 is to perform a critical review of existing approaches in automatic temporal processing. Such an overview is useful for assessing the positive aspects and the drawbacks characterising existing methods in order to contextualise and justify choices made throughout this research. It is also necessary for identifying the relevant contributions brought to the domain.

Goal 3 is to build, annotate and investigate corpora and resources which are used throughout this research. The type of annotation applied is guided by the type of information that needs to be identified. The different corpora provide both statistical data concerning the frequency of different phenomena in text, as well as a basis for system evaluation.

Goal 4 is to design, implement and evaluate the methodology concerned with identifying the textual extent of temporal expressions. This process relies on distinguishing different knowledge sources that can prove useful, implementing them as separate modules, and then evaluating them as part of the system to allow an analysis of their impact on system performance.

Goal 5 is to determine the best methodology for the disambiguation of the temporal adverb *then*, an adverb of great communicative strength that easily expresses various semantic categories. At the temporal expression identification stage it is important to distinguish the anaphoric usage of *then* that realises the semantic role of time, as only these occurrences of *then* need to be considered temporal expressions.

Goal 6 is to find the best approach for the normalisation of temporal expressions, a process that involves finding the value that a certain expression designates or is intended to designate. This requires experimenting with different normalisation models, and dealing with various problems that appear at this stage. The implementation of these models and solutions to the problems posed by the normalisation process should be done in such a way that the resulted modules are highly customisable, so that a comprehensive evaluation can be performed. A comparative evaluation of the system results when integrating each module should help choose the best solution to the normalisation problem.

Goal 7 is to develop and evaluate a method for the identification and annotation of events expressed using verbs in natural language texts. Achieving this goal requires access to a resource that assigns to each verb an event class that best characterises that verb.

Goal 8 is to design and evaluate a methodology for the identification of temporal relations between events and temporal expressions, or between events and other events.

Goal 9 is to automatically identify temporal clauses by disambiguating the ambiguous subordinators that can introduce them.

Goal 10 is to identify the limitations of the proposed methodology, and to propose ways forward.

1.5 Outline of the thesis

This thesis is organised into eight chapters which approach temporal processing gradually: from theoretical foundations (Chapter 2) to practical methods (Chapter 3), continuing with the original contribution in the areas of temporal expression identification (Chapter 4) and normalisation (Chapter 5); event annotation (Chapter 6); temporal relation identification (Chapter 7), and finishing with conclusions (Chapter 8). Each chapter addresses one or more research goals.

Chapter 2 discusses various theoretical issues involved in the area of temporal processing. It focuses on the mechanisms used by language to express temporal information, and on the linguistic efforts made to formalise them. This chapter serves as a theoretical foundation for this research, and addresses **Goal 1**.

Chapter 3 presents existing temporal annotation schemes, resources, and computational approaches employed so far to perform different temporal processing tasks. The benefits and drawbacks of each approach are carefully examined to determine which is the best methodology that should be used in this work for solving each task. This chapter accomplishes **Goal 2**.

Chapters 4 to 7 concentrate on the original contributions of this research.

Chapter 4 addresses the first stage involved in the task of temporal expression annotation. This stage is known as temporal expression identification and deals with detecting the textual extent of the temporal expressions present in a given text. Considering the example presented in Section 1.1, this stage corresponds to Task 1. The chapter starts with an exhaustive classification of the most common types of TEs encountered in natural language texts. The TE identifier is developed so that it can reliably identify all the TE types captured in the classification. However, certain temporal expressions require additional attention from a module that checks the syntactic correctness of an identified TE, or from a module that identifies when the adverb *then* is used anaphorically and should be considered a temporal expression. The accurate recognition of a particular usage of *then* relies on the investigation of an annotated resource that captures the different semantic categories expressed by *then*. The development of this resource addresses **Goal 3**. This chapter also contains a comparative evaluation that illustrates the improvement brought by each module to the overall system performance. The system is first evaluated for its ability to annotate according to the TIMEX2 annotation scheme (Ferro et al., 2005), and afterwards it is adapted to the TIMEX3 annotation scheme (Pustejovsky et al., 2003), followed by another evaluation. This chapter contributes to **Goals 3, 4 and 5**.

Chapter 5 describes the second stage of the temporal expression annotation process that deals with the normalisation of temporal expressions (Task 2 exemplified in Section 1.1). At this stage the values of the attributes assigned to a TE are identified. These attribute values can either be extracted from

the expression itself, or calculated using the attribute values of another TE which serves as an anchor. Several tracking models can be envisaged for finding the most appropriate anchor for an under-specified TE. This work experiments with four temporal anchor tracking models, all having different levels of context dependency. It also addresses two important problems that influence the quality of the normalisation process: **the direction problem** and **the generic vs. specific problem**. A comparative evaluation of the four temporal anchor tracking models is also included in this chapter, along with evaluations of the system including the modules solving the two normalisation problems mentioned above. These evaluations focus on the TIMEX2 system produced annotation. After adapting the system to the TIMEX3 standard, another evaluation is performed. This chapter addresses **Goal 6**.

Chapter 6 focuses on the identification and classification of events denoted by either finite or non-finite verbs. The event identification process relies entirely on the information provided by the syntactic parser. The classification problem is more complicated, due to its semantic nature. This problem is solved by annotating each verb present in WordNet 2.0 (Fellbaum, 1998) with the event class that is most suitable for the verb's meanings. The resulting resource is employed in the annotation of verbal events with TimeML-compliant (Pustejovsky et al., 2003) information. This method is then evaluated by comparing the system output with the gold standard annotation. The work presented in this chapter addresses **Goals 3 and 7**, and solves Task 3 illustrated in Section 1.1.

Chapter 7 proposes a novel methodology for discovering temporal relations

that hold among events and temporal expressions (Task 4 exemplified in Section 1.1). Language uses several mechanisms to encode temporal relations. Temporal clauses are an important mechanism that has not been investigated in the previous chapters. To overcome this, the identification of temporal clauses is addressed in this chapter by first compiling and annotating a corpus of temporal clauses, and then adopting a machine learning method that detects when ambiguous subordinators are used to introduce temporal clauses. Now that the system can identify the most important mechanisms used by language to express temporal relations, this information is exploited with the aim of automatically identifying the temporal relation between two temporal entities. To this end, a novel methodology that relies on the propagation of temporal relations in syntactic trees is proposed and evaluated. This chapter accomplishes **Goals 3, 8 and 9**.

Finally, **Goal 10** is achieved in the last chapter of the thesis. **Chapter 8** summarises the contributions of this research, discusses how the goals of this thesis have been fulfilled, and identifies potential future directions of research.

Chapter 2

Time in natural language

2.1 Overview

This dissertation is motivated by the intention to capture the temporality of a given text. It is thus appropriate to begin by investigating how time is perceived by humans. Section 2.2 provides a short overview of the different perspectives from which time can be understood. This chapter then describes how time is expressed in natural language and represents an account of time-related issues from a theoretical perspective. English, as well as any other natural language, possesses several mechanisms for expressing temporal information which can broadly be grouped into three large categories: temporal expressions, events and the temporal relations between temporal expressions and/or events. The most important temporal mechanisms are described in detail in the following sections, grouped under three headings that correspond to the three major types of temporal information: temporal expressions in Section 2.3, events in Section 2.4, and temporal relations in Section 2.5.

2.2 What is time and how is it conveyed in text?

The *Oxford Dictionary of English* (Soanes and Stevenson, 2005) defines **time** as "a limited stretch or space of continued existence, as the interval between two successive events or acts, or the period through which an action, condition, or state continues".

Time has been a major subject of controversy in religion, philosophy, and science, and a definition of time applicable to all fields of study is unlikely to be adopted. For example, some philosophers view time as part of the fundamental structure of the universe, a dimension in which events occur in sequence (Rynasiewicz, 2004). Newton believed time and space form a container for events which is as real as the objects it contains. In contrast to Newton's belief in absolute time and space, Kant (1999) considers that time does not refer to any kind of container that events and objects "move through", nor to any entity that "flows", but is instead part of a fundamental intellectual structure within which humans sequence and compare events. McTaggart (1908) speaks of time as "temporal becoming" or events changing from being future, to being present, to being past. Aristotle (350 BC) [1] answered the question "What is time?" by declaring that "time is the measure of change", while emphasising "that time is not change [itself]" because a change "may be faster or slower, but not time". This is now referred to as the relational theory of time.

All these opinions seem to agree on the one-way direction of the so-called **arrow of time** pointing from past to future, and on the fact that time provides a baseline reference point in which events can be placed in order of occurrence: in this manner people can establish that one event occurred before or after another.

1. http://classics.mit.edu/Aristotle/metaphysics.html

To gain computational insights into the mechanisms that build this arrow of time in natural language, an investigation of how language is used to convey temporal information is necessary. Insights resulting from such studies may help to formulate assumptions and hypotheses that can then be exploited to automatically "understand" the temporality of a given text. Even though the linguistic data analysed throughout this thesis is limited to English, the interpretative principles formulated here should apply to other natural languages.

English, like any other natural language, possesses several mechanisms for expressing temporal information which can broadly be grouped into three large categories: temporal expressions, events and the temporal relations that hold among times and events. The most important temporal mechanisms are described in detail in the following sections, grouped under three headings that correspond to the three major types of temporal information: temporal expressions (Section 2.3), events (Section 2.4), and temporal relations (Section 2.5).

2.3 Temporal expressions

Temporal expressions (referred to throughout this thesis as **time expressions** or **TEs**) are natural language phrases that refer directly to time, giving information about when something happened, how long something lasted, or how often something occurred. The way temporal expressions are lexicalised in natural language is the subject of Section 2.3.1.

Time expressions denote calendar dates, times of day, periods of time, durations or sets of recurring times. Most temporal expressions in English play the syntactic role of circumstance adverbials that express the semantic role of time. Temporal expressions convey different types of time-related information:

position, duration, frequency and relationship (Biber et al., 1999; Quirk et al., 1985). Sections 2.3.2, 2.3.3 and 2.3.4 describe how position in time, duration and frequency, respectively, are expressed in natural language. Time expressions can also indicate temporal relationship. This will be discussed in more detail in Section 2.5.

2.3.1 Grammatical realisation of temporal expressions

Temporal expressions possess a wide range of grammatical realisations. Especially notable is the use of noun phrases that appear either individually or preceded by a preposition and take the form of prepositional phrases (PPs). It should be noted that prepositions preceding time expressions are not included in the extent of the TE, but since they indicate temporal relationships, they will be discussed in more detail in Section 2.5. Noun phrases that include subordinate relative clauses which determine nouns denoting temporal concepts, such as *period*, *week* (e.g. *the week that he was away*), are also considered to be temporal expressions. Another frequent way to designate time expressions is represented by certain adverbs, adjectives or their corresponding phrases (adverbial or adjectival phrases).

A temporal expression is usually signalled by one or more time words, called **lexical triggers**, such as:

- nouns: *century, year, month, day, weekend, minute, future, past*;
- proper names: *Christmas, April, Sunday*;
- adjectives: *past, current, future, next, medieval, monthly*;
- adverbs: *currently, then, weekly, today, yesterday, tomorrow, tonight*;

- specialised time patterns: *9:00, 26/12/2002, '80s*;
- numbers: *4th* (as in *John arrived on the 4th.*).

2.3.2 Expressing position in time

Temporal expressions can indicate position in time, specifying when something takes place, and typically serving as a response to a potential *When* question. Whenever a time position is expressed using noun phrases, it frequently includes determiners, such as *that* in example [2.1]. However, noun phrases cannot be normally used to express a pinpointed time position, and in such cases, prepositional phrases are more appropriate (example [2.2]). Time position can also be expressed using adverbs: the most frequent ones according to Biber et al. (1999) being *now, then, today, ago, yesterday* (example [2.3]).

[2.1] *Mary met him* ***that afternoon****.*

[2.2] *The wedding was on* ***Thursday****.*

[2.3] *John went for a walk* ***yesterday****.*

Time position can either be precisely indicated by the use of time points (example [2.4]), or vaguely specified by expressions which delimitate time periods or intervals (example [2.5]). The following sentences extracted from Allen (1983) exemplify the two usages:

[2.4] *We found the letter at* ***twelve noon****.*

[2.5] *We found the letter* ***yesterday****.*

The temporal expression *twelve noon* introduced by the preposition *at* in example [2.4] indicates a precise time point at which the letter was found, while the temporal expression *yesterday* from example [2.5] refers to a temporal interval in which the finding of the letter occurred.

2.3.3 Expressing temporal duration

Another meaning carried by time expressions is duration, in which case they represent appropriate answers to *How long* questions. Durations offer the greatest freedom to use noun phrases (example [2.6]), though in most cases these noun phrases can be regarded as abbreviated prepositional phrases lacking the preposition *for* (example [2.7]). Time expressions denoting duration are typically formed by adjoining a quantifier (e.g. *several*, *three*, *many*) with a time unit (e.g. *year*, *week*, *hour*).

[2.6] *They lived* ***several years*** *in Italy.*

[2.7] *His mother-in-law stayed (for)* ***three weeks****.*

2.3.4 Expressing frequency in time

Temporal expressions can also convey frequency, describing how often something occurs. Such expressions of frequency with respect to a specified or implied span of time can normally represent answers to *How often* questions. For expressing frequency, noun phrases usually have the construction *every/each + T* (example [2.8]), where *T* is either a time unit (e.g. *hour, day*) or another word referring to time (e.g. *Monday*), but time units and other temporal words can also appear as bare plurals without any determiner (example [2.9]). Prepositions like *on*, *at* or *in* can combine successfully with noun phrases to express time frequency (example [2.10]). Another way to express frequency is provided by adjectives and adverbs derived from time units (e.g. *hourly*, *monthly*, *annually*) (example [2.11]).

[2.8] *Mary writes an article or a review* ***every month****.*

[2.9] ***Saturdays*** *John goes to the theatre.*

[2.10] *They reviewed their stock portfolio on* ***the first day of each month****.*

[2.11] *A* ***monthly*** *newsletter is emailed to all customers.*

Along with temporal expressions, events constitute another important temporal phenomenon that greatly contributes to the temporal information of a given text. The theoretical aspects involved in the temporal treatment of events are discussed in the following section.

2.4 Events

Natural language sentences or clauses describe what some call **eventualities** (Bach, 1986), and others call **situations** (Comrie, 1976; Smith, 1991). **Eventuality** is a cover term for **states** and **events**, introduced by Bach (1986). A **state** is an eventuality in which there is no relevant change during the span of time over which the state is true (e.g. *know someone*, *being happy*). An **event** is an eventuality that involves a change of state (e.g. *learn a language*, *build a house*). Events can be seen as dynamic situations that imply change and/or movement, and they include actions initiated by agents.

In the literature, the terminology concerned with eventualities or situations, events and states is inconsistently used, a fact also acknowledged by Tenny and Pustejovsky (2000), who label this terminology related to events as "unstable". Several classes of eventualities have been distinguished by different authors, but the typical distinction is the one made between **non-statives** (events) and **statives** (states) (Vendler, 1967; Dowty, 1979; Bach, 1981; Jackendoff, 1990; Verkuyl, 1993; Pustejovsky, 1995; Rappaport Hovav and Levin, 1998). This distinction appears to be cognitively basic from the point of view of change,

as events involve a change from an initial state to a resulting one (e.g. *build*), while states denote properties or relations that do not change throughout the spans of time over which the states hold (e.g. *love*) (Dowty, 1979; Parsons, 1990; Pustejovsky, 1995). However, much recent work including Briscoe et al. (1990) and Pustejovsky (1995) has adopted the term **event** to express what was originally conveyed by the term **eventuality** introduced by Bach (1986). The same perspective is assumed in this thesis. In the following, unless clearly stated, the term **event** refers to what is otherwise included under the term **eventuality**, and therefore it also stands for **states**.

Not only is the terminology describing events inconsistent, but it has also been impossible for researchers to agree what represents the extent of an event. Different approaches have considered events as being expressed using several types of text units, including lexical items like verbs (example [2.12]), nouns (example [2.13]) or adjectives (example [2.14]), verb phrases (example [2.15]), clauses (example [2.16]), sentences (example [2.17]), and semantic entities (example [2.18]). As a result, different distinctions can be made in the same sentence as to what is the number and extent of the events it contains. For example, sentence [2.17] can be seen by some authors as one single event, while other authors would see it as embedding two events, one denoted by the verb *rejected* and a second one designated by the noun *offer*.

[2.12] *In fiscal 1989, Elco* ***earned*** *$7.8 million, or $1.65 a share.*

[2.13] *Ms. Atimadi says the* ***war*** *has created a nation of widows.*

[2.14] *They say IRA commanders are* ***responsible*** *for the recent bomb attacks.*

[2.15] *Rally's Inc. said it* ***has adopted a shareholders rights plan****.*

[2.16] *The Federal Bureau of Investigation says* ***it received more than eight thousand reports of hate group crimes last year****.*

[2.17] ***Telerate's two independent directors have rejected the offer.***

[2.18] *We know that 3,000 teens start smoking each day,* ***although it is a fact that 90% of them once thought that smoking was something that they'd never do.***

However, most linguists associate events with the tensed verb that is central to a sentence or a clause, and by extension with that sentence or clause. This is the reason why event analysis is normally centred on properties of the verb, and justifies the choice of verbal events as representative of the event class in the context of this thesis. This strong correlation between verbs and events is validated by numerous efforts to classify verbs according to how the events they denote take place in time. Semantically, this temporal internal contour of an event is captured by the notion of **lexical aspect** (Rothstein, 2004).

2.4.1 Lexical aspect

Lexical aspect is the inherent property of an eventuality concerned with the manner in which that eventuality develops or holds in time. This notion is deployed to classify eventualities into different categories according to their temporal semantics. In the literature, it is also referred to as **Aktionsart** (Agrell, 1908), **semantic aspect** (Comrie, 1976), **aspectual class** (Dowty, 1979), **situation type** (Smith, 1991), or **eventuality type** (Bach, 1986).

The category of lexical aspect has been traditionally distinguished from the aspectual properties introduced by grammaticalised morphemes such as the perfective or imperfective verbal morphology found in many languages. The aspectual properties expressed by a grammatical category or characterised by a particular inflectional morphology determine the **grammatical aspect** of the verb. This is the category one normally refers to when mentioning the

term **aspect**: "aspect in linguistic terminology is usually understood to refer to different inflectional affixes, tenses, or other syntactic ≪frames≫ that verbs can acquire (aspect markers)", according to Dowty (1979). Dowty recognises that "semantic differences inherent in the meanings of verbs themselves cause them to have differing interpretations when combined with these aspect markers, and that certain of these kinds of verbs are restricted in the aspect markers and time adverbials they may occur with". For instance, when combining a non-stative verb like *sing* with the progressive aspect, the resulting construction is stative (e.g. *Mary was singing*). The "semantic differences inherent in the meanings of verbs themselves" Dowty mentions, refer to the notion of lexical aspect and contribute to distinguishing the aspectual class of a verb. Dowty relies on the fact that certain classes of verbs may occur only with a restricted set of grammatical aspect markers, in justifying the use of the term aspect in a wider sense to apply also to the aspectual classes of verbs. Several criteria are used in determining aspectual verb classes, and these are detailed below.

2.4.2 Criteria for aspectual classification

Three basic semantic dimensions are normally used as criteria for classifying events into aspectual categories: **dynamicity**, **durativity**, and **telicity** (Comrie, 1976).

Dynamicity is the most basic aspectual notion setting apart events that involve change, also called **non-stative** or **dynamic events** (example [2.19]), from the ones that do not involve change, also known as **statives** or **states** (example [2.20]).

[2.19] *Mary walked to the shop.*

[2.20] *John loves his job.*

Durativity distinguishes between **instantaneous events** that take place at a point in time (example [2.21]), and **durative events** that last a certain amount of time (example [2.22]). This differentiation between events that "occur at a single moment" vs. events that "last for a period of time" (Vendler, 1967) is also present in the literature as the **punctuality** vs. **temporal extension** distinction (Moens and Steedman, 1988), or as **the indivisibility property** (Bach, 1986).

[2.21] *Mary won the dancing contest.*

[2.22] *John slept during the contest.*

Telicity is the property of an event to have an end point or to be directed towards a goal. According to this feature events can be **telic** denoting movements toward an end point or a culmination, or **atelic**. The distinction between telic and atelic dates back to Aristotle (350 BC), who first observed that some verb meanings necessarily involve an end or result in a way that others do not. He distinguished between **kinesis**, translated as **movements**, indicating actions that are directed toward an end (example [2.23]), and **energeia**, translated as **actualities**, referring to actions that are complete in themselves (example [2.24]). Telic events are therefore equivalent to Aristotelian kinesis, while atelic ones correspond to Aristotelian energeia.

[2.23] *John fixed the roof.*

[2.24] *Mary was happy to be home.*

Multiple terms are used in the literature to capture the telic vs. atelic distinction: **bounded** vs. **non-bounded** (Verkuyl, 1993), **culminating** vs. **non-culminating** (Moens and Steedman, 1988), **delimited** vs. **non-delimited** (Tenny, 1987), or **definite** vs. **indefinite change of state** (Dowty, 1979).

2.4.3 Aspectual categories

Much work on lexical aspect relies on the aspectual categories initially introduced by Vendler (1967), even if, over the years, refinements and alterations to his typology in terms of lexical and syntactic categories involved, linguistic tests, and semantic formalisation have been advanced by various authors including Dowty (1979), Bach (1986), Moens and Steedman (1988), Smith (1991), and ter Meulen (1995). As a result, various classification systems have been proposed, though they make essentially the same distinctions, collapsing some classes or subdividing others. Each of these classifications builds on previous ones in an attempt to provide a formalised way of distinguishing aspectual categories. From the point of view of Computational Linguistics this is very important because, without a reliable way of distinguishing these categories by humans, it is impossible to implement automatic systems able to identify them. Possessing the ability to discriminate between these aspectual categories is an extremely important step towards temporally understanding a text, as each category is characterised by different temporal properties. The distinctions and observations made by the above authors have guided the definition of the temporal annotation standard TimeML (Pustejovsky et al., 2003) and much of the work involved in automatically identifying event classes.

The classification systems for aspectual categories are described in more detail below, starting with Vendler's work, and continuing in chronological order with other relevant contributions to the definition and formalisation of aspectual class typologies, such as the proposals of Dowty, Bach, Moens and Steedman, Smith and ter Meulen.

Vendler

Following traditional Aristotelian classes, Vendler (1967) laid out a typology of events underlying verb uses, and marked the beginning of this tradition in lexical semantics literature. Vendler identified four aspectual verb classes based on temporal properties such as temporal duration, temporal termination, and internal temporal structure. In the Vendler classification, verbs may denote **states**, **activities**, **achievements** or **accomplishments**. Each of them is detailed in the following paragraphs.

States have no internal temporal structure: they last for a period of time, and they involve no change during the span of time over which they are true (example [2.25]). Vendler argues that states lack continuous tenses, at the same time acknowledging that verbs which are clearly states in their dominant usage can sometimes be used with progressive tenses to refer to an activity (see the definition of an activity below). To illustrate this, he gives the example of the verb *think* in two different contexts: one where the verb is used with a continuous tense and refers to an activity (example [2.26]), and another one reflecting the most common use of the verb *think* as a state (example [2.27]).

[2.25] *Mary loves art.* (state)

[2.26] *John is thinking about Mary.* (activity)

[2.27] *Mary thinks that rabbits are cute.* (state)

Activities (or **processes**) are ongoing events with internal change and duration, but no necessary temporal end point, that consist of successive phases following one another in time (e.g. considering the *running* event from example [2.28], the man who is running lifts up his right leg one moment, drops it the next, then

lifts his other leg, drops it, and so on). They are characterised by **temporal homogeneity**, i.e. the property of an event of taking place at a given interval as well as at any subpart of this interval (Dowty, 1986). Therefore, if it is true that someone has been running for half an hour, then it must be true that he has been running for every period within that half hour.

[2.28] *John is running.*(activity)

Accomplishments are events which have duration and a definite end point (example [2.29]). Vendler observes that while the event of *drawing* (as in example [2.30]) has no set terminal point, *drawing a circle* does have a "climax" or, in other words, it culminates. He points out that accomplishments, like activities, go on in time, but, unlike activities, they proceed toward a terminus, thus lacking temporal homogeneity (i.e. if someone has drawn a circle in two minutes, it cannot be true that he has drawn a circle in any period included in those two minutes).

[2.29] *Mary is drawing a circle.*(accomplishment)

[2.30] *Mary is drawing.*(activity)

Achievements have an instantaneous culmination, lacking duration (example [2.31]). Since achievements do not extend over time, they typically do not allow temporal *for*-adverbials and lack the ability to be used with continuous tenses. As in the case of states, a change of aspectual class occurs when using a continuous tense with certain verbs which generally denote achievements. For example, by combining the verb *win* with a progressive tense as in example [2.32], its aspectual class changes from achievement to activity (process), the resulting construction refers to the process by which the *winning* achievement was obtained. This is

due to the progressive auxiliary requiring its argument to be a process that it describes as ongoing.

[2.31] *John won the race.*(achievement)

[2.32] *John was winning the race at that point.*(activity)

Vendler claims that, in the vast majority of cases, verbs fall completely, or at least in their dominant use, within one of the four delimited classes, thus assuming that the verb determines the aspectual class. The same view was adopted in this thesis by associating an aspectual class with each English verb, in an attempt to solve the event classification task described in detail in Chapter 6. However, many other authors (Dowty, 1979; Tenny, 1987; Thompson, 2005) promote the view that aspectual properties belong to the verb phrase or the clause, rather than to the verb itself. This is due to many factors, including adverbial modification, the influence of the verb's arguments, as well as grammatical aspect. The fact that grammatical aspect influences the aspectual class of a verb phrase or clause was illustrated above by combining progressive tenses with verbs generally describing states or achievements. This contextually determined change of aspectual class is known as **aspectual composition**.

Dowty

The classification proposed by Vendler has the drawback of relying on very few examples, which makes it difficult to assimilate. To compensate for this, Dowty (1979) proposed an informal list of different verbs/verb phrases that correspond to each class, as well as several syntactic and semantic tests to identify members of each aspectual class. Some examples of verbs/verb phrases proposed by Dowty as instances of Vendler's four categories are:

- **States**: *know*, *believe*, *have*, *desire*;
- **Activities**: *run*, *walk*, *swim*, *drive a car*;
- **Accomplishments**: *paint a picture*, *make a chair*, *draw a circle*;
- **Achievements**: *spot*, *find*, *lose*, *reach*, *die*.

The collection of tests for aspectual classification recommended by Dowty is summarised in Table 2.1.

For example, Dowty uses the following adverbial test for the telic vs. atelic distinction: temporal adverbial expressions introduced by the preposition *in* modify sentences representing bounded (telic) events (example [2.33]), while temporal adverbial expressions introduced by *for* modify non-bounded (atelic) events (example [2.34]).

[2.33] *John built the house* ***in one year/*for one year.***(telic) [2]

[2.34] *John danced* ****in ten minutes/for ten minutes.***(atelic)

One test useful for distinguishing accomplishments from other event types relies on the fact that only accomplishments can be found as complements of the verb *finish*. This particular verb requires that its complement describe an event that involves both a process and a culmination.

[2.35] *Mary finished writing the letter.*(accomplishment)

[2.36] **John finished building.*(activity) [3]

[2.37] **Mary finished spotting John.*(achievement)

[2.38] **John finished knowing Mary.*(state)

2. Throughout this thesis, the symbol * will be used to indicate that certain propositions are either anomalous or highly unlikely to be expressed in natural language.

3. This sentence is acceptable in cases of object ellipsis (i.e. previous context gives information about what John was building), and in such cases it expresses an accomplishment.

Criterion	**States**	**Activities**	**Accomplishments**	**Achievements**
1. meets non-stative tests	no	yes	yes	?
2. has habitual interpretation in simple present tense	no	yes	yes	yes
3. *V for an hour,* *spend an hour Ving*	OK	OK	OK	bad
4. *V in an hour,* *take an hour to V*	bad	bad	OK	OK
5. *V for an hour* entails *V at all times in the hour*	yes	yes	no	d.n.a.
6. *X is Ving* entails *X has Ved*	d.n.a.	yes	no	d.n.a.
7. complement of *stop*	OK	OK	OK	bad
8. complement of *finish*	bad	bad	OK	bad
9. ambiguity with *almost*	no	no	yes	no
10. *X Ved in an hour* entails *X was Ving during that hour*	d.n.a.	d.n.a.	yes	no
11. occurs with *studiously*, etc. *attentively, carefully*, etc.	bad	OK	OK	bad

OK = the sentence is grammatical, semantically normal
bad = the sentence is ungrammatical, semantically anomalous
d.n.a. = the test does not apply to verbs of this class
yes = verbs of this class pass the test
no = verbs of this class do not pass the test
? = achievements are like statives according to some stativity tests, but not others

Table 2.1: Dowty's tests for aspectual verb categories (from Dowty (1979))

It is worthwhile mentioning that these linguistic tests can vary in reliability. Dowty himself observed that the syntactic tests for distinguishing Vendler's categories fail to give consistent results. To address this, he then introduces more criteria for classifying events based on agentivity (referring to the existence of an agent that carries out the action denoted by the verb) and on the distinction between complex vs. simple change of state (referring to whether a change of state can or cannot be considered to consist of two or more temporally consecutive subsidiary changes). With this refined classification, he tries to justify certain inconsistencies encountered in Vendler's classification, as well as in the results of the tests proposed for aspectual class delimitation. The fact that these refinements do not throw more light on how to recognise members of a certain aspectual class, but rather add more complexity and ambiguity to this process, motivates the choice of not presenting them in more detail here.

Bach

Bach (1986) introduced the notion of eventuality and proposed the division of eventualities into states and non-states, capturing a distinction imposed by the notion of change, a distinction that is deemphasised in Vendler's and Dowty's classifications. The aspectual classes distinguished by Bach are presented in Figure 2.1.

Bach distinguishes between two kinds of states according to their ability to occur with progressive tenses: dynamic states and static states. Only dynamic state verbs can freely occur with progressive tenses. Dynamic states are episodic, they apply only to spatio-temporal slices of individuals (example [2.39]). Static states hold permanently of their arguments (example [2.40]), or can be predicated of

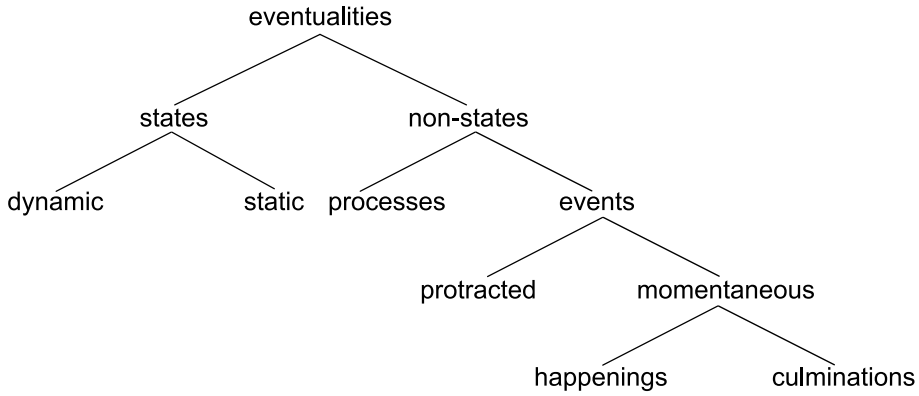

Figure 2.1: Aspectual classes distinguished by Bach

them atemporally (example [2.41]).

[2.39] *Mary is feeling sick.*

[2.40] *John knows the answer./*John is knowing the answer.*

[2.41] *The earth is round./*The earth is being round.*

Non-states are further subdivided into processes (equivalent to Vendler's activities) and events (subsuming Vendler's accomplishments and achievements). Events are protracted (Vendler's accomplishments) or momentaneous (Vendler's achievements). Momentaneous events are split into culminations (example [2.42]) and happenings (example [2.43]), according to whether they involve a transition to a new state that is associated with culminations, but not with happenings.

[2.42] *John's father died a few years ago.*(culmination)

[2.43] *Mary noticed John's mistake.*(happening)

Bach (1981) also tries to elucidate the parallel between the mass-count distinction in nominal systems and the process-event distinction in aspectual classifications of verbal expressions. He uses the mereological **part-of** relation to establish the analogy between the nominal pair **things-stuff** and its verbal

correspondent **events-processes**. For example, there is a similar mapping between a thing (e.g. *ring*) and the stuff it is made out of (e.g. *gold*), and telic events (example [2.44]) and the process stuff they are made out of (example [2.45]). In the same way that things have boundaries that delimit them in space, events have boundaries that delimit them in time, while stuff and processes either do not have boundaries, or when they do the boundaries are vague, unknown or irrelevant.

[2.44] *Mary drank a glass of wine.*(event)

[2.45] *Mary drank wine.*(process)

Moens and Steedman

Moens and Steedman (1988) extend Vendler's work and introduce another class of events called **points** that are instantaneous and involve no culmination, a class also encountered in Bach's categorisation as happenings. The authors distinguish the following aspectual types: states, processes, culminated processes, points and culminations. They delimit these classes on the grounds of durativity (**atomic** vs. **extended** events), and association with a consequent state (**+consequent state** vs. **-consequent state**). This subcategorisation is captured in Table 2.2, taken from Moens and Steedman (1988). Even if the authors preserve most of the classes defined by Vendler unchanged, they modify the nomenclature to avoid any confusion caused by the old terms. They want to highlight the fact that Vendler's accomplishments, which they call **culminated processes**, are "composite events, consisting of a process which is associated with a particular culmination point" (Moens and Steedman, 1988).

States are, according to their definition, "indefinitely extending states of affairs"

	EVENTS		STATES
	atomic	**extended**	
	CULMINATION	**CULMINATED PROCESS**	
+conseq	*recognize*	*build a house*	*understand*
	spot	*walk a mile*	*love*
	win the race	*eat a sandwich*	*know*
	POINT	**PROCESS**	*resemble*
-conseq	*hiccup*	*run*	
	tap	*swim*	
	wink	*play the piano*	

Table 2.2: Subcategorisation of event types proposed by Moens and Steedman

(example [2.25], page 27). Moens and Steedman preserve the distinction between events and states imposed by the notion of change.

Processes are defined as events that extend in time which are not characterised by any conclusion or culmination (example [2.28], page 28). These non-conclusive events determine the class denoted by the term activity in Vendler's typology.

Culminated processes represent durative processes which culminate and cause a change of state, being previously termed accomplishments or protracted events (example [2.29], page 28).

Points are events that are viewed "as an indivisible whole and whose consequences are not an issue in the discourse" (Moens and Steedman, 1988). They sound odd in combination with perfect tenses (example [2.46]), probably because the perfective grammatical aspect class, whenever combined with the

present tense, typically indicates that a certain action that occurred at a certain point in the past has consequences in the present.

[2.46] **Harry has hiccupped.*

Culminations are punctual or instantaneous events, which are accompanied by a transition to a new state, called the consequent state of the event. Both points and culminations are included in Vendler's achievement class. Culminations combine with perfect tenses, the resulted statement emphasising the corresponding consequent state (example [2.47]).

[2.47] *Harry has reached the top.*

Smith

Smith (1991) uses the term situation to refer to what Bach (1981) called eventuality. Smith distinguishes five types of situations: states, activities, accomplishments, semelfactives and achievements. They differ in the temporal properties of dynamicity, durativity, and telicity.

States are "stable situations which hold for a moment or an interval" (Smith, 1991). They have the temporal features of being static and durative. States have no dynamics, and include "the ascription of concrete and abstract properties of all kinds, possession, location, belief and other mental states, dispositions, habits, etc" (example [2.25]).

Activities are defined as "processes that involve physical or mental activity, and consist entirely in the process" (Smith, 1991). They are dynamic, durative and atelic, they refer to situations of gradual change, and they do not require that a particular degree is reached. This class is equivalent to the activity class

delimited by Vendler (example [2.28]).

Accomplishments "include process and outcome" (Smith, 1991). They are dynamic, durative and telic, corresponding to the same class in Vendler's typology (example [2.29]).

Semelfactives are "single-stage events with no result or outcome" (Smith, 1991). They are characterised by the dynamic, atelic and instantaneous features (e.g. *knock at the door*, *hiccup*, *flap a wing*). They normally occur very quickly, with no outcome or result other than the occurrence of the event.

Achievements are defined as "instantaneous events that result in a change of state" (Smith, 1991). They have the dynamic, telic and instantaneous properties (e.g. *win a race*, *reach the top*, *leave the house*).

ter Meulen

ter Meulen (1995) claims that aspect "controls the dynamics of the flow of information about described change encoded in a text". Besides states, ter Meulen distinguishes three means of dynamic "flow control": **holes**, **filters**, and **plugs**, corresponding to the three traditional aspectual classes of events.

Holes correspond to what previous authors have called activities or processes (example [2.28]), that is "events that apply throughout their internal structure homogeneously" (ter Meulen, 1995).

Filters correspond to accomplishments or culminated processes (example [2.29]), and are defined as "descriptions of change that never apply to any part of an event they describe" (ter Meulen, 1995).

Plugs are "special cases of filters, commonly called «achievements»(example [2.31]), which are in a conceptual sense instantaneous, since they do not consist of an initial and a final stage" (ter Meulen, 1995).

The author then illustrates how the flow of information is controlled by holes, filters and plugs. Given an event describing a hole, the information conveyed by the following sentence can be seen as if it was flowing through the hole or, in other words, as being a temporal part of the hole event. An event that describes a filter restricts the information flowing through it to be interpreted as either denoting a later event or denoting an event temporally included in the filter. If a clause describes an event as a plug, then it blocks all information about anything happening at the same time. A plug event forces the next sentence to be interpreted as describing a later event, thus redirecting the temporal focus. This is due to the fact that a plug is seen as an instantaneous or atomic event constrained in such a way that it has no temporal parts accessible for future description.

Having introduced the most relevant classifications of events into aspectual categories, one can now conclude that the features distinguished by most authors are dynamicity and telicity, and to these some authors add further refinements (durativity, occurring with progressive, etc). Table 2.3 summarises the most important aspectual classification systems proposed so far in the literature, with a view towards providing a general picture of existing aspectual classes. This table illustrates that the aspectual classification systems proposed so far make essentially the same distinctions, despite collapsing, subdividing or giving different names to certain categories.

The ability to distinguish between aspectual classes is crucial to determining

Vendler	Bach	Moens and Steedman	Smith	ter Meulen
States	***Dynamic states***	***States***	***States***	***States***
"may be predicated of a subject for a given time with truth or falsity"	expressed by stative verbs that have the ability to occur with progressive tenses	"indefinitely extending states of affairs"	"stable situations which hold for a moment or an interval"	"do not have any internal temporal structure"
	Static states			
	hold permanently of their arguments or can be predicted of them atemporally			
Activities	***Processes***	***Processes***	***Activities***	***Holes***
"processes going on in time [...] in a homogeneous way"	"we think easily about a process going on for a time and think about smaller chunks of the same process"	"describe an event as extended in time but not characterised by any particular conclusion or culmination"	"processes that involve physical or mental activity, and consist entirely in the process"	"events that apply throughout their internal structure homogeneously"
Accomplishments	***Protracted events***	***Culminated processes***	***Accomplishments***	***Filters***
"they also go on in time, but they proceed toward a terminus which is logically necessary to their being what they are"	"correspond to Vendler's accomplishments"	"a state of affairs that also extends in time but that does have a particular culmination associated with it at which a change of state takes place"	"include process and outcome"	"descriptions of change that never apply to any part of an event they describe"
Achievements	***Culminations***	***Culminations***	***Achievements***	***Plugs***
"occur at a single moment"	are momentaneous events like "die", "reach the top"	"an event which the speaker views as punctual or instantaneous"	"instantaneous events that result in a change of state"	are "special cases of filters, commonly called «achievements», which are in a conceptual sense instantaneous, since they do not consist of an initial and a final stage"
	Happenings	***Points***	***Semelfactives***	
	are momentaneous events like "notice", "recognise"	"event [...] that is viewed as an indivisible whole and whose consequences are not an issue in the discourse"	"single-stage events with no result or outcome"	

Table 2.3: Summary of aspectual classification systems

the correct temporal interpretation of a given text. Considering for instance the problem of finding the temporal order between two successive sentences, the aspectual classes of the two main events play a crucial role in deciding what is the temporal order of the two events in time. Knowing the aspectual classes of the two main events, one can decide the temporal order of the two sentences on the basis of the following principles (Dowty, 1986):

a. "If a sentence in a narrative contains an accomplishment or achievement predicate but no definite time adverb, that sentence is understood to describe an event occurring later than the time of the previous sentence's event".

b. "If on the other hand the second sentence of the sequence has a stative predicate [...] or an activity predicate [...], the state or process it describes is most usually understood to overlap with that of the previous sentence".

2.5 Temporal relations

Temporal relations are relations that hold between temporal entities, i.e. between events, between an event and a TE, and between two TEs. A temporal relation is "an inter-propositional relation that communicates the simultaneity or ordering in time of events or states" (Longacre, 1983).

On the basis of what is explicitly uttered as having happened, people automatically make all kinds of inferences about what must have happened when, and about what the exact succession of events was. Some of these inferences are immediately enabled by the information explicitly present in what was said, others require more reasoning to uncover what is rather left implicit.

This section explores how temporal relations are conveyed in English, focusing mostly on phenomena that are automatically identifiable and that will be

exploited in the development of an automatic system targeting the identification of temporal relations.

It is natural to start by describing the set of temporal relations widely used by researchers to capture the temporal dimension of a narrative. Section 2.5.1 presents this set of 13 temporal relations distinguished by Allen (1983). Allen's temporal relations have been commonly adopted by the research community, although due to their high specificity more and more researchers confronted with practical annotation issues are working with sub-sets of the original set of 13 relations.

After presenting the set of temporal relations one can encounter in natural language, the following sections examine the mechanisms one can employ to infer the temporal relations present in each utterance: time adverbials, tense, grammatical aspect, as well as other implicit ways to express temporal relations.

2.5.1 Allen's theory

To be able to reason about time, efficient ways of representing temporal entities and the relations between them are needed. Amongst the most influential work in this area is that of Allen (1983, 1984, 1991). Allen considers that every event can be seen as having a start point and an end point that define a temporal interval taken by that event on the timeline. He also considers that TEs can be mapped to temporal intervals (for example *today* can be represented by the temporal interval [2008-12-01T00:00 [4], 2008-12-01T23:59]).

Considering that both events and temporal expressions can be mapped to intervals, Allen has identified 13 possible **interval - interval** temporal relations.

4. This representation of dates and times is defined by the ISO 8601 international standard covering *Data elements and interchange formats – Information interchange – Representation of dates and times.*

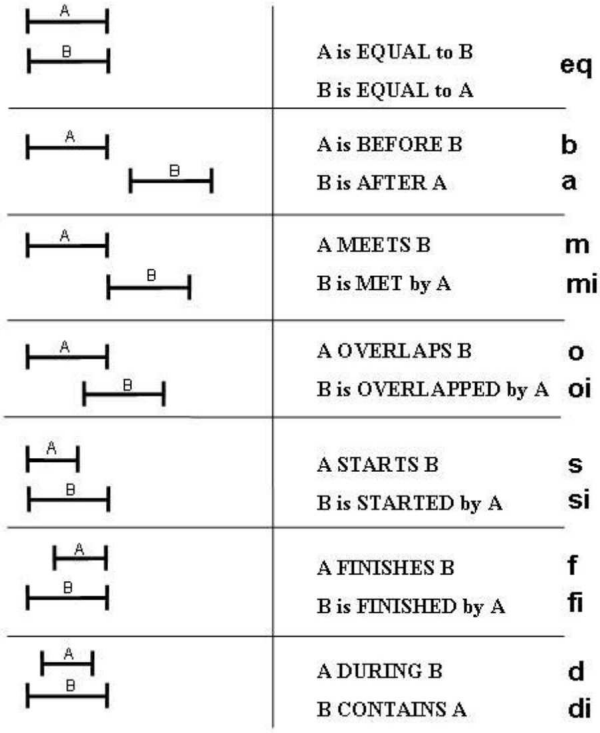

Figure 2.2: Allen's set of temporal relations

One relation is the **identity relation (eq)** between two intervals, six relations are **before (b)**, **meets (m)**, **overlaps (o)**, **starts (s)**, **finishes (f)**, **during (d)**, and the other six are their inverses: **after (a)**, **is met by (mi)**, **is overlapped by (oi)**, **is started by (si)**, **is finished by (fi)**, **contains (di)**. All 13 relations are explained in Figure 2.2.

After reducing all events and temporal expressions to intervals and after identifying the temporal relations between them, the temporal information in a text can be represented as a graph where events and TEs form the nodes, and the edges are labelled with the temporal relations between them. Figure 2.3 illustrates a time graph representing the temporal information included in the following news article that was given as an example in Setzer (2001):

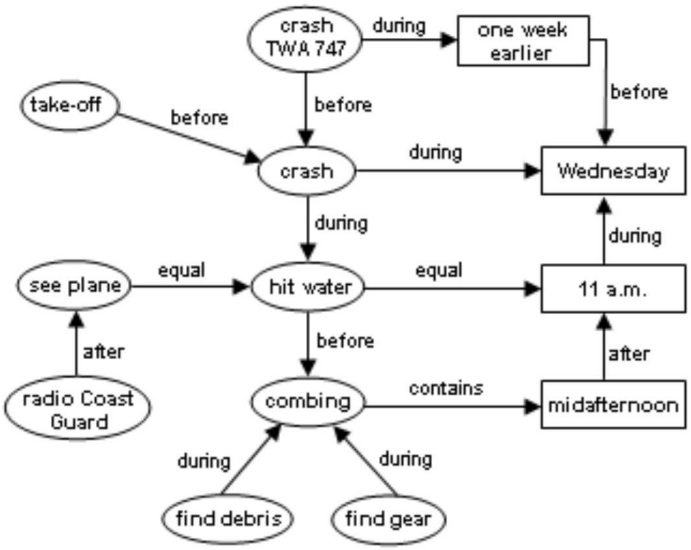

Figure 2.3: Example of a time graph

Small plane crashes into Atlantic; no survivors found

A small single-engine plane crashed into the Atlantic Ocean about eight miles off New Jersey on Wednesday. The Coast Guard reported finding aircraft debris and a fuel slick, but no bodies or survivors. The plane, which can carry four people, was seen hitting the water shortly after 11 a.m. by a fisherman, who radioed the Coast Guard, according to Petty Officer Jeff Fenn, a spokesman for the base at Governors Island in New York Harbor. By midafternoon, several vessels and a helicopter were combing the area about eight miles east of Sea Bright, N.J., and seven miles south of the Ambrose Light, the Coast Guard said. The area is 55 miles from the site off Long Island where a TWA 747 crashed one week earlier. Searchers found the plane's landing gear, seat cushions and other debris, Petty Officer Fenn said. He said the water is about 125 feet deep in the crash area and that much of the wreckage had sunk. The Coast Guard said the craft had taken off from Allaire Airport in Monmouth County, N.J. The Federal Aviation Administration said the plane was registered to Delaware Environmental Development Services of Wilmington. There was no listing for the company in Wilmington.

The main problem is that natural languages do not usually express directly the interval which a given event takes on the timeline in terms of its specific

start and end points. Temporal relations are typically only partially expressed in natural language, via several mechanisms presented below.

2.5.2 Tense

Tense is a specific mechanism built into language for locating information in time. It can be defined as the "grammaticalized expression of location in time" (Comrie, 1976). Tense usually refers to the ability of verbs to change form in order to convey information about the location of an event in time. For example, in [2.48], the past tense morpheme *-ed* generating the inflected form of the verb *to dance* is used to indicate that the event occurred at a time earlier than the time of the utterance (also known as the **speech time**). In [2.49], the modal auxiliary *will* is used to locate the event as occurring at a future time with respect to the speech time.

[2.48] *Mary* ***danced*** *at the party.*

[2.49] *Mary* ***will dance*** *at the party.*

Tense is typically marked by an inflection of the verb using suffixes like null morpheme/*-(e)s* for the present tense, and the suffix *-ed* for the past tense. The existence of a future tense is argued by some grammarians, while others claim that the future tense does not exist, as tense is a category strictly realised by verb inflection. Morphologically English has no future form of the verb, it merely expresses the semantic category of **future time** via certain grammatical constructions such as *will* + infinitive. This thesis acknowledges these opinions, but for ease of presentation will use the term **Future Tense** to refer to verbal constructions expressing future time. Adhering to any of the above positions would make no difference to the methodology adopted in this research.

In sequences of adjacent sentences or coordinated clauses, tense is an

extremely important source of information at one's disposal when identifying the temporal relations between two events.

When looking at the semantics of each individual tense, one notices that the name of a tense does not necessarily capture the time of the event expressed by the inflected verb. In the following the meaning of each individual tense is described in detail, according to the information presented in Quirk et al. (1985).

Present Tense

The present tense is the most general category, giving away little information about the time of a situation. It can have the following usages:

- **Timeless present:** in this case the present tense is used without reference to specific time, mostly in statements expressing so-called **eternal truths** ([2.50]);

[2.50] *Water* ***consists*** *of hydrogen and oxygen.*

- **Habitual present:** refers to an event that repeats itself over a period of time ([2.51]);

[2.51] *They* ***visit*** *their parents every week.*

- **Instantaneous present:** occurs when a verb is used to refer to an action that was begun and completed approximately at the moment of speech ([2.52]);

[2.52] *I* ***advise*** *you to quit.*

- **Simple present referring to the past:** describes the past as if it would be happening in the present. It is also called the **historic present**. ([2.53])

[2.53] *Just as John arrived, Mary* ***leaves*** *the room.*

- **Simple present referring to the future:** is typically encountered in main clauses accompanied by a time adverbial locating the action in the future ([2.54]), or in subordinate conditional and temporal clauses ([2.55]).

[2.54] *The airplane* ***departs*** *at 9pm tomorrow.*

[2.55] *John will call when he* ***receives*** *the book.*

Past Tense

Past tense usually expresses the fact that the event took place in the past, and that there is a gap between its completion and the speech time.

The use of past tense can be anaphoric, in the sense that "its interpretation is linked to some time or event derived from context" (Webber, 1988). This phenomenon appears in contexts where the time of a past tense event is interpreted with respect to either a time expressed by a temporal adverbial in the same sentence, or an event described in previous discourse. In example [2.56], the time when John saw many squirrels is to be interpreted relative to the given context of his going to the park. The clause *John went to the park* just creates a temporal background within which the event described by the clause *and saw many squirrels* is to be located.

[2.56] *John went to the park and* ***saw*** *many squirrels.*

Future Tense

Future tense usually refers to a time after the speech time. The most important constructions used for expressing future time include:

- the modal auxiliary construction with *will*, *shall*, or the contracted form *'ll* followed by the infinitive form of the verb;

- *be going to* followed by the infinitive;

- simple present;

- *be to* or *be about to* followed by the infinitive.

The grammatical category of tense, together with the lexical aspect described

in Section 2.4.1 and the grammatical aspect explained in the following section, are extremely relevant for this thesis not only in the context of the work on temporal relations presented in Chapter 7, but also as important features used to characterise events in Chapter 6.

2.5.3 Grammatical aspect

Aspect, or more precisely **grammatical aspect**, refers to how a certain situation is viewed by the speaker with respect to time, i.e. whether it is conceived as completed (**perfective**) or ongoing (**imperfective** or **progressive**). This is why it is often referred to in the literature as **viewpoint aspect**.

Grammatical aspect represents a formal distinction encoded in the grammar of a language. The perfective aspect is syntactically realised using the auxiliary *to have* followed by the past participle form of the verb (e.g. *have eaten*), while the progressive aspect is signalled by the auxiliary verb *to be* followed by the *-ing* participle form of the verb.

Tense and aspect combine freely in the complex verb phrase, and they are very closely connected in meaning. The most usual use of the word **tense** is to refer to a combination of what we have described above as tense and grammatical aspect (e.g. *Past Perfect*, *Past Progressive*, *Simple Past*). While the names **Perfective** and **Progressive** are used to illustrate which of the two categories of grammatical aspect is present in that verb phrase, the name **Simple** describes a verb phrase totally unmarked for aspect.

The overlap of meaning between tense and aspect is most problematic in English when choosing between Simple Past and Present Perfective. Both [2.57] and [2.58] indicate a state of affairs that took place before the present moment, but the Simple Past indicates that the period of two years has ended, whereas the

Present Perfective indicates that the residence has continued up to the present time and may even continue into the future. The Present Perfective can be seen as representing past time with present relevance and through its usage it is implied that the event is still relevant at the time of speaking.

[2.57] *John lived in London for two years.* (Simple Past)

[2.58] *John has lived in London for two years.* (Present Perfective)

The perfective aspect defines an anterior time zone within which the action of the verb takes place. This anterior time zone precedes whatever time orientation is signalled by tense or by other elements of the sentence or its context.

The progressive aspect indicates an event in progress at a given time. The sentences [2.59] and [2.60] are identical in terms of location in time, as they both locate the situation in the past, but they differ in terms of aspect in the sense that the former statement describes the event as a whole, while the latter makes reference to an ongoing event.

[2.59] *Mary danced.*

[2.60] *Mary was dancing.*

The progressive aspect signals that Mary's dancing is a temporary and not a permanent phenomenon and that the event took place over a period of time, rather than happening all at once.

The category of lexical aspect introduced in Section 2.4.1 (also known as Aktionsart) is different from the grammatical aspect this section focuses on. In this case, the aspectual properties are introduced by grammaticalised morphemes such as the perfective or imperfective verbal morphology found in many languages. Unlike Aktionsarten that are related to inherent lexical properties of verbs or verb phrases, grammatical aspect operates more in the syntactic domain. Both lexical aspect and grammatical aspect are important

features heavily exploited in the work concerning events (Chapter 6) and temporal relations (Chapter 7).

2.5.4 Reichenbach

One of the most influential pieces of work aiming at a deep understanding of how temporal relations are encoded in text is the work of Reichenbach (1947). Reichenbach argues that utterances marked for tense and aspect introduce references to three time points: **the speech time S**, **the event time E**, and **the reference time R**. The speech time is the time at which the utterance is produced. The event time is the time at which the described event occurred. The reference time is the time from which the speaker is viewing the event on a timeline.

The difference between these three time points is illustrated in [2.61]. In this example, the event time is the time when John returned from his holiday, the reference time is *Sunday*, the time by which John's return had already taken place, and the speech time is the time at which the sentence is uttered.

[2.61] *On Sunday John had already returned from his holiday.*

Three temporal relations can hold between these time points: **at** or =, **before** or <, and **after** or >. In example [2.62], the event time (i.e. the time of reading the book) is situated before the reference time (i.e. the time when John told her the plot), and the reference time is situated before the speech time (this is indicated by the past tense of the verb *told* that locates the action in the past with respect to the speech time). A simple temporal illustration of this example would be $\mathbf{E} < \mathbf{R} < \mathbf{S}$.

[2.62] *<S>Mary had [read the book]<E> [when John told her about the plot]<R>.*

In terms of Reichenbach's theory, the relation between the reference time R and the speech time S is established by tense, while the relation between the event time E and the reference time R is provided by grammatical aspect. Therefore, for present tense the reference time coincides with the speech time (R = S), for past tense the reference time is situated before the speech time (R < S), while for future tense the inverse temporal relation applies (R > S). As far as grammatical aspect is concerned, the relations established between the event and the reference time are as follows: for the simple aspect the two times coincide (E = R), while for the perfective aspect the time of the event is located on the timeline before the reference time (E < R).

Reichenbach's analysis of the English tense – aspect system is illustrated in figure 2.4.

One can easily notice in figure 2.4 that the progressive/continuous tenses are treated by Reichenbach as the event time having extended intervals in time instead of time points, but are otherwise similar to the simple tenses.

This theory is an important source of guidance to understanding how tense and aspect contribute to the temporal ordering of events, being extremely relevant to the approach adopted in Chapter 7 for temporal relation identification.

2.5.5 Time adverbials

Temporal relations are especially dependent for their expression upon time adverbials. Time adverbials convey temporal relations between the time they denote and the verbal event they syntactically depend on.

Time adverbials are syntactically realised by means of adverbs, noun phrases, prepositional phrases and temporal clauses. Most adverbs (e.g. *yesterday*), noun phrases (e.g. *last week*) and prepositional phrases (e.g. *on Monday*) that

Relation	Reichenbach's Tense Name	English Tense Name	Example
E < R < S	Anterior Past	Past Perfect	I had slept.
E < R < S	Anterior Past Extended	Past Perfect Progressive	I had been sleeping.
E = R < S	Simple Past	Simple Past	I slept.
E = R < S	Simple Past Extended	Past Progressive	I was sleeping.
E < S = R	Anterior Present	Present Perfect	I have slept.
E < S = R	Anterior Present Extended	Present Perfect Progressive	I have been sleeping.
S = R = E	Simple Present	Simple Present	I sleep.
S = R = E	Simple Present Extended	Present Progressive	I am sleeping.
S < E < R	Anterior Future	Future Perfect	I will have slept.
S < E < R	Anterior Future Extended	Future Perfect Progressive	I will have been sleeping.
S < R = E	Simple Future	Simple Future	I will sleep.
S < R = E	Simple Future Extended	Future Progressive	I will be sleeping.

Figure 2.4: Reichenbach's interpretation of the English tense – aspect system

express the semantic role of time are considered temporal expressions, and it should be noted that temporal expressions form the largest subclass of time adverbials. Temporal clauses (e.g. *John came home* ***after Mary left***.) are another realisation of time adjuncts, a temporal clause being able to relate the time of the event it mentions to the time of the event described in the clause it syntactically depends on. While temporal expressions relate an event to a time, temporal clauses establish temporal relations between two events. The time of the event described in the main clause may be previous to, subsequent to, or

simultaneous with the time of the event described by the temporal clause.

Section 2.3 has already described in detail the different types of time-related information TEs can convey: position, duration, frequency and relationship. These subroles also apply to all time adverbials, as one can see in their description below.

Time position adverbials

When expressing time position, time adverbials can narrowly pinpoint the exact time an event took place ([2.63]), or they can denote a wider time interval to which the event time belongs ([2.64]).

[2.63] *Mary left* ***at 10:30 am****.*

[2.64] *John went to India* ***last year****.*

In both cases, time position adverbials refer to a span of time within which, at some point of time, the events took place. This applies to prepositional phrases introduced by *on*, *at* or *in*.

Prepositional phrases introduced by *after* or *before* place the time of the event denoted by the verb after, respectively before, the time denoted by the noun phrase following these prepositions.

Time position can also be expressed via temporal clauses. Temporal clauses introduced by *when* and *as* indicate the simultaneity of the events in the main and subordinate clauses, whereas subordinators like *after*, *as soon as* and *once* indicate that the event expressed in the main clause takes place after the event of the subordinate clause. The opposite effect is obtained by temporal clauses introduced by *before*, as they indicate that the event in the main clause happened before the one in the temporal clause.

Durative adverbials

Temporal duration can either be expressed as a general temporal measure, or the duration can be anchored either to a specific position on the time axis or to the time of an event.

Durations expressed using noun phrases and prepositional phrases introduced by the prepositions *for* indicate the fact that an event lasted the exact amount of time denoted by the durative expression.

Durations expressed by prepositional phrases introduced by *during*, *within*, *over*, *throughout* indicate that the time of the event is included in the specified time span. Similarly, temporal clauses introduced by *while* and *whilst* indicate that the time of the event in the main clause is included in the time span denoted by the durative clause they introduce.

In the case of the temporal subordinators *as long as* and *so long as*, both the main clause and the subordinate clauses are durative and these subordinators generally indicate that the situations begin and end at the same time, thus emphasising both simultaneity and duration.

Less specificity is encountered in the case of durative prepositional phrases introduced by prepositions like *until*, *till*, *up to*, and *to*.[5] Such PPs indicate a time interval extending from a reference time point prominent in discourse to the point in time specified by the phrase in question. The temporal relation suggested in this case is that the event is bound by the specified temporal interval. The same applies to temporal clauses introduced by *until* and *till*, with the only difference consisting in the end point of the interval being specified as being the time of the event mentioned in the *until*/*till*-clause.

5. This statement applies to PPs introduced by *to* only when they are correlated with *from*-PPs (e.g. *from December 1998 to June 2005*).

Durative PPs introduced by the prepositions *since* and *from* indicate an interval starting from the point in time specified by the PP and ending at the reference time that is prominent at that point in the discourse. Temporal clauses introduced by *since* are in a similar position, indicating that the interval starts at the moment in time indicated by the *since*-clause.

Frequency adverbials

Time adverbials can also convey frequency, describing how often an event occurs. They are mostly realised by adverbs (e.g. *weekly*), noun phrases (e.g. *every day*), or temporal clauses introduced by *whenever* and in certain cases by *when*.

Frequency adverbials can express definite frequency (e.g. *annually*) or indefinite frequency (e.g. *usually*), but in all cases they indicate a repetitive nature of an event with either a specified or unspecified frequency.

Temporal clauses introduced by *whenever* or *when* (when used to imply repetitiveness) may imply that the events of the main and of the subordinate clause overlap in time if at least one of the clauses is durative ([2.65]).

[2.65] *Mary is careful* ***whenever she crosses a street.***

Time relationship adverbials

Time adverbials can express a relationship between two time positions that are both being considered in an utterance. They are typically realised by adverbs that signal temporal sequence, such as: *afterwards*, *then*, *before*, *later*, *next*, *previously*, *subsequently*. They indicate the temporal relation that holds between the event expressed by the verb they syntactically depend on, and the reference time point or the event that was last introduced in the preceding discourse.

This section has shown that time adverbials represent an important source of information in the process of inferring the temporal relations holding between events and temporal expressions. Since in most cases time adverbials are expressed using temporal expressions, the relations they typically provide are between the event expressed by the verb the adverbial depends on, and the temporal expression forming the adverbial. In the case of noun phrases and most adverbs in the role of a time adverbial, the temporal relation conveyed is one of simultaneity. For example, given the sentence [2.66], the event *moved* and the time adverbial *last year* are overlapping temporally. In the case of prepositional phrases and temporal clauses, the temporal relation is typically indicated by the preposition or the subordinator. In example [2.67] the subordinator *after* indicates that the event of moving is temporally after the event of graduating.

[2.66] *Mary* ***moved*** *to France* ***last year****.*

[2.67] *Mary* ***moved*** *to France* ***after*** *she* ***graduated****.*

In this thesis, all types of time adverbials will be automatically identified using first a methodology to identify time expressions (see Section 4.3 for more details), and then a different methodology targeting the identification of temporal clauses which is described in detail in Section 7.2.

2.5.6 Other ways of expressing temporal relations

Besides the mechanisms described so far, one can encounter other ways in which language expresses temporal relations.

At the syntactic level, temporal relations can be inferred by examining certain dependency relations. For example the temporal expressions included in noun phrases to qualify the noun heading the NP indicate that there is a temporal relation of overlap between the event denoted by the head of the noun phrase

and the time indicated by the temporal expression (see [2.68]).

[2.68] *They do not know the result of* ***the Sunday election****.*

At the semantic level, an important role is played by world knowledge. Without world knowledge it is often impossible to know that an event represents an integrating part of another event, or that an event causes another event.

One semantic factor that is capable of encoding temporal relations appears in the case of **subevents**. In example [2.69], the fact that the event of painting the walls from example [2.69] is part of redecorating the house leads to the interpretation that the temporal relation between the two events is one of temporal inclusion.

[2.69] *John* ***redecorated*** *his house. He first* ***painted*** *the walls.*

Causality is another factor that intervenes at the semantic level and is also dependent on world knowledge. If an event causes the occurrence of another event, then the temporal relation holding between the two events is one of temporal precedence, as the cause always comes before the effect. In [2.70], despite the fact that the event of pushing is mentioned after the event of falling, it is located in time before the falling event.

[2.70] *John fell. Mary pushed him.*

Temporal relations can also be expressed by **narrative sequence**. In example [2.71], one naturally understands that the event of going home happened before cooking dinner which was before eating it. The sequence in which these events appear in text reflects the order in which they happened.

[2.71] *Mary went home. She cooked dinner and ate it in front of the TV.*

In a given text events can be mentioned several times, and this leads to the phenomenon of **event co-reference**. The referential instance of an event takes place at the same time the event that serves as the antecedent does, and one can

infer from here that all the temporal relations holding for one instance of the event also holds for the other one.

This brings about an important source of temporal relations: **inference**. Temporal relations can be inferred using simple rules, such as the transitivity rule: if an event A happens before an event B, and B happens before C, then one can easily infer that A happens before C.

This is an insight into how temporal relations can be found in text and gives an idea of how complex the entire process of identifying temporal relations in text would be. While temporal relations made explicit in text via mechanisms such as tense, grammatical aspect or temporal adverbials can be automatically identified, the semantically implicit temporal relations presented in this section pose real challenges to automatic systems due to the world knowledge required for their identification.

2.6 Conclusions

This chapter described mechanisms used by natural language to convey temporal expressions, events and temporal relations. It focused on the most common phenomena used by language to express and conceptualise time with a view towards employing this knowledge in the development of an automatic system that would be able to identify temporal expressions, events and temporal relations in text. Despite trying to make this analysis as comprehensive as possible, due to language variability it is impossible to cover all existing ways to express time in natural language.

The survey targeted in turn the three major types of temporal information: temporal expressions in Section 2.3, events in Section 2.4, and temporal relations in Section 2.5. The knowledge included in this chapter will be used in the next chapters to propose automatic methods for extracting different types of temporal information.

Chapter 3

Computational approaches and existing resources for temporal processing

3.1 Overview

This chapter provides an overview of existing resources and computational approaches used for the identification of temporal expressions, events and temporal relations in news articles, most relying on the theoretical framework presented in Chapter 2. The current chapter does not attempt to present an exhaustive survey of the existing methods. Instead it focuses on the most important approaches and systems that are chosen either as typical instances of classes of systems or methods, or because they represent a notable advance on previous work, or because the approach they take is interesting and original. A preference is shown towards implemented systems over theoretical proposals, and towards methods which are, or might be made to be, applicable to a wide range of tasks against highly task-oriented methods.

Since annotating temporal information in text would be impossible without defining annotation standards, the chapter starts by describing existing temporal

annotation schemes in Section 3.2. These annotation schemes guided the annotation of several resources with temporal information markup, and Section 3.3 describes the most relevant resources for this research.

Section 3.4 describes the main approaches taken towards the identification and normalisation of temporal expressions. The computational treatment of events is captured in Section 3.5, and Section 3.6 focuses on how temporal relations have been addressed in the literature.

3.2 Annotation schemes

This section presents chronologically the annotation schemes that have been extensively used in the past for the development of resources for temporal processing.

3.2.1 The first TIMEX

The first annotation scheme that encoded temporal information was developed for the MUC (Message Understanding Conferences) campaigns. MUC was a series of evaluation exercises that aimed to measure the performance of Message Understanding (MU) systems, now referred to as Information Extraction (IE) systems (Sundheim and Chinchor, 1993). These evaluation exercises included several tasks, such as the task of Named Entity Recognition (NER). The NER task required the identification and classification of different types of named entities such as: persons, locations, organisations, dates, times and monetary values.

Dates and times were first included as targeted classes of named entities in MUC-5 (Sundheim, 1993), and they were present until the last MUC conference,

MUC-7 (Chinchor, 1998). Time expressions were supposed to be identified in text and annotated using the SGML tag TIMEX. This tag was characterised by only one attribute, TYPE, that captured the type of the temporal expression and took either the value DATE for date expressions, or TIME for expressions denoting times of the day.

As far as time expressions were concerned, the MUC NER tasks tested the accuracy of systems in recognising TE extent, and they did not require resolution of the TE values. The identification of temporal expressions was only a step towards filling the slots of different scenario templates. Scenario template filling required the identification of specific relations between template elements, in this case between times and events. Participating systems were required to assign a time to certain event types. For example, in the case of rocket launch events, the scenario template contained a field called LAUNCH_DATE that was linked to the appropriate time entity. The temporal relation between the time and the event was not further evaluated. Temporal relations between events and other events were not addressed.

After the last MUC in 1998, the Automatic Content Extraction (ACE) campaigns (ACE, 1999) replaced the MUC exercises and increased the complexity of the tasks. In ACE more temporal expressions were targeted, and the annotation involved the highly complex TIMEX2 tag, described in more detail below.

3.2.2 The TIDES TIMEX2

The TIDES TIMEX2 is an annotation scheme for marking the extent of English time expressions and representing their values according to the ISO-8601 (ISO8601:2004, 2004) standard format. It was developed to support research

activities under the DARPA TIDES (Translingual Information Detection, Extraction and Summarisation) research program (TIDES, 2002), and the Automatic Content Extraction program (ACE, 1999).

The TIMEX2 annotation scheme extends the MUC-7 scheme by widening the range of markable expressions, and by replacing the TIMEX TYPE attribute with a set of attributes that specify in more detail the semantic representation of a time expression. In addition, the TIDES TIMEX2 scheme is compliant in terms of the format used to represent time values with the ISO-8601 standard.

TIMEX2 was originally developed during the year 2000 under the TIDES program, and was first documented in Ferro et al. (2000). It has then undergone several revisions yielding newer versions of the guidelines described in Ferro et al. (2001, 2003), with the latest version being presented in Ferro et al. (2005).

The latest annotation guidelines describe a wide set of markable time expressions, including mostly the temporal expressions presented in Section 2.3. According to the guidelines, the full extent of a TE should either be a noun, adjective, adverb or any of the corresponding phrases (noun, adjectival or adverbial phrases). The temporal expression cannot be a prepositional phrase or a clause, so it cannot start with a preposition or a subordinating conjunction (e.g. *after* ***Friday*** and *before they meet on* ***Monday*** are disallowed as temporal expressions, only ***Friday*** and ***Monday*** being correct markables). Premodifiers of temporal expressions such as determiners, and postmodifiers such as prepositional phrases or subordinate clauses should be included in the time expression. The appositives that may appear after a TE are not to be included in the expression's tag, but, if they contain temporal trigger words, they are to be tagged separately.

In the case of temporal range expressions (e.g. *from* ***1990*** *to* ***1999***), and

POINTS IN TIME	
VAL= "YYYY-MM-DDThh:mm:ss" <TIMEX2 VAL="2004-02-23T15:00">3 p.m. Monday</TIMEX2>	Anchored expressions T = ISO time-of-day designator
VAL="YYYY-WOY-D" <TIMEX2 VAL="2004-W10">next week</TIMEX2>	Week-based format
VAL="token" <TIMEX2 VAL="PRESENT_REF">now</TIMEX2>	Tokens that replace the entire value of VAL
VAL="YYYY-*token*" <TIMEX2 VAL="2003-FA">Fall 2003</TIMEX2> **VAL="YYYY-MM-DDT*token*"** <TIMEX2 VAL="2004-02-24TMO">Tuesday morning</TIMEX2> **VAL="WOY-*token*"** <TIMEX2 VAL="W09-WE">this weekend</TIMEX2>	Tokens that replace particular positions in the value of VAL
DURATIONS	
VAL="PnYnMnDTnHnMnS" <TIMEX2 VAL="P1H">one hour long</TIMEX2> **VAL="PnW"** <TIMEX2 VAL="P3W">three weeks</TIMEX2>	Expressions answering the question *how long*

Table 3.1: Possible formats of the TIMEX2 attribute VAL

conjunction (e.g. ***today*** *and* ***tomorrow morning***) or disjunction (e.g. ***six months*** *or* ***a year from now***) of time expressions, the points should be tagged separately, even if they share modifiers.

The tag element used to mark up time expressions is TIMEX2, and its attributes are: VAL, MOD, ANCHOR_VAL, ANCHOR_DIR, SET and COMMENT. The TIMEX2 tag attributes are presented below together with their use.

The VAL attribute is used for any expression that indicates a point or interval on a calendar/clock or that can be identified as an unanchored duration. The placeholder character "X" is used when parts of the value are unknown. The possible formats of VAL are captured in Table 3.1.

The value of VAL can include certain tokens relevant in the representation of time points and durations that can occupy the entire value of VAL, or tokens covering only parts of the value. These tokens are listed in Table 3.2.

The MOD attribute is used together with other attributes when the time expression includes a modifier that changes or clarifies the interpretation of VAL in some way. MOD captures the semantics of quantifier modifiers (e.g.

TOKENS COVERING THE WHOLE VALUE OF *VAL*		
Token	**Markable expressions**	**Non-markable expressions**
PAST_REF	past yesterday former lately long ago medieval	before previously earlier beforehand once
PRESENT_REF	now today current, currently present, presently nowadays (at) this (point in) time (at) the present time (at) the present moment	immediately instantly forthwith
FUTURE_REF	future tomorrow	ahead after soon, sooner shortly later eventually subsequent
TOKENS OCCUPYING ONLY ONE POSITION IN *VAL*		
Token	**Expressions**	**Position**
MO MI AF DT EV NI	morning midday afternoon daytime or working hours evening night	Hour
WE	weekend	Day
SP SU FA WI Qn H1 H2	spring summer fall, autumn, fall term/semester winter n-th quarter (n = 1..4) first half (of year) second half (of year)	Month

Table 3.2: Tokens that may appear in the value of the TIMEX2 attribute VAL

approximately, no more than) and lexicalized aspect markers (e.g. *early, start of*), but not the semantics of prepositions or other terms outside the temporal expression. The tokens representing possible values for MOD, together with expressions that trigger them are presented in Table 3.3.

The attributes ANCHOR_VAL and ANCHOR_DIR are always used together to indicate the orientation and anchoring of certain durations with respect to other points or periods of time. The value of the ANCHOR_VAL attribute is the normalisation of the anchoring date or time in ISO format, while the value of the ANCHOR_DIR attribute shows the orientation of the duration with respect to the date or time denoted by ANCHOR_VAL. The possible values of ANCHOR_DIR are: WITHIN, STARTING, ENDING, AS_OF, BEFORE, AFTER. For example,

TYPE OF EXPRESSIONS	VALUES OF *MOD*	EXPRESSIONS
POINTS IN TIME	BEFORE AFTER ON_OR_BEFORE ON_OR_AFTER	more than ... ago less than ... ago no less than ... ago no more than ... ago
DURATIONS	LESS_THAN MORE_THAN EQUAL_OR_LESS EQUAL_OR_MORE	less than ... (long), nearly more than ... (long) no more than at least
POINTS AND DURATIONS	START MID END APPROX	early, dawn, start, beginning middle, mid- end, late about, around, approximately

Table 3.3: Tokens that may represent the value of the TIMEX2 attribute MOD

given the expression *the three months ending May 31*, ANCHOR_VAL would be assigned the value *2010-05-31*, and ANCHOR_DIR the value *ENDING*.

The SET attribute is used in the representation of expressions denoting sets of time, i.e., times that recur regularly or irregularly (e.g. *every Tuesday, numerous weeks, some Thursdays*) and its only value is YES.

The COMMENT attribute was introduced so that annotators can insert remarks about why they made a specific decision for ambiguous expressions, or to signal certain cases of doubt.

The TIMEX2 annotation guidelines are the most refined annotation specifications developed so far for any temporal entity, therefore the resulting annotated corpus described in detail in Section 3.3.1 is very reliable. In addition, this also enables the development of automatic systems achieving good performance for the task of TIMEX2 annotation (Section 4.4 provides more details of the results obtained by automatic systems performing TIMEX2 annotation). However, the TIMEX2 annotation scheme is concerned only with time expressions, and to be able to build a temporal representation of a given text one needs ways to represent not only temporal expressions, but also the information related to events and temporal relations holding among temporal expressions and events. This need is addressed by STAG, an annotation scheme

that enables the annotation of the three most important temporal phenomena - temporal expressions, events and temporal relations - in a given text.

3.2.3 STAG

STAG (Sheffield Temporal Annotation Guidelines) is the temporal annotation language presented in Andrea Setzer's PhD thesis (Setzer, 2001). Setzer proposes an annotation scheme which enables time expressions, events and temporal relations to be marked up in newswire texts. The resulting annotation scheme is briefly described below by looking at how each type of temporal entity should be annotated.

Annotating time expressions

The STAG annotation scheme distinguishes two types of time expressions: **simple** (*last Thursday*) and **complex** (*17 seconds after hearing the sound*).

Simple time expressions are those expressed using adverbs and noun phrases that do not contain a reference to an event as part of the noun phrase. Simple time expressions should be annotated by marking their entire text span.

Complex time expressions arise when an event is syntactically dependent on the head noun of the TE, and their value should be interpreted with respect to the time of the subordinated event.

Both simple and complex time expressions should be annotated using the SGML tag <TIMEX>, which has the following attributes:

- **tid**: the ID that uniquely identifies the time expression in the text;
- **type**: the type of the time expression (possible values: DATE, TIME, COMPLEX);

- **calDate**: the calendar date represented by the expression, in the format [[DD]MM]YYYY or (SPR|SUM|AUT|WIN)YYYY;
- **eid**: the ID of the event the time expression is related to;
- **signalID**: the ID of the signal indicating the temporal relation between the event and the time expression;
- **relType**: the temporal relation holding between the time expression and the event (possible values: BEFORE, AFTER, INCLUDES, IS_INCLUDED, SIMULTANEOUS)

The attributes eID, signalID and relType apply only to complex time expressions and they are used to give information about the relation of type relType that holds between the time expression and the event eid, relation established via the signal signalID.

Annotating events

STAG considers markable events to be the head of the finite verb group expressing an event, the head of the noun phrase for events expressed using nominalisations, and the non-finite verb in the case of an event expressed in a non-finite clause. Events are annotated using the SGML tag <EVENT> that is characterised by the following attributes:

- **eid**: the event ID that uniquely identifies the event in text;
- **class**: one of the following classes that an event can belong to: OCCURRENCE, PERCEPTION, REPORTING, ASPECTUAL;
- **argEvent**: the id of the argument event usually taken by reporting, perception and aspectual events;

- **tense**: shows whether the event happens in the past, present or future (potential values: PAST, PRESENT or FUTURE);
- **aspect**: illustrates the grammatical aspect of the verb, and therefore it can receive one of the following values: PROGRESSIVE or PERFECTIVE;
- **relatedToEvent**: the ID of the event that the current event is temporally related to;
- **eventRelType**: the type of the temporal relation holding between the two related events (possible values: BEFORE, AFTER, INCLUDES, IS_INCLUDED, SIMULTANEOUS);
- **relatedToTime**: the ID of the time expression the current event is related to;
- **timeRelType**: the type of temporal relation holding between the event and the time expression (possible values: BEFORE, AFTER, INCLUDES, IS_INCLUDED, SIMULTANEOUS);
- **signalID**: the ID of the text span that signals the temporal relation between two entities.

Annotating temporal relations

Events can be related to time expressions or to other events. In the case of an event related to a time expression, the ID of the time expression and the temporal relation between the two entities are stored in the attributes relatedToTime and timeRelType included in the SGML tag of the event. To annotate event–event relations, the event ID of one event is stored as a value of the attribute relatedToEvent in the SGML tag of the other event. The temporal relation between the two is stored in the attribute eventRelType. If either of the two types of temporal relations is explicitly signalled, then the ID of the signal is stored in the attribute signalID.

The concepts included in the STAG and TIMEX2 annotation schemes are integrated together in a more general-purpose specification language for tagging all three types of temporal phenomena: TimeML.

3.2.4 TimeML and ISO-TimeML

TimeML (Pustejovsky et al., 2003; Saurí et al., 2006) is a formal specification language for events, temporal expressions and their orderings, developed as a result of a wide interest in temporal analysis and event-based reasoning. This interest was manifested in a number of important specialised workshops and satellite events organised at major conferences including ACL 2001 (ACL-2001, 2001), LREC 2002 (LREC-2002, 2002), TERQAS 2002 (TERQAS, 2002), TANGO 2003 (TANGO, 2003), Dagstuhl 2005 (Dagstuhl, 2005), TIME 2006 (TIME-2006, 2006), ARTE 2006 (ARTE, 2006). Significant progress was made during these events, leading to the design and refinement of TimeML.

Compared to its predecessors, TimeML is a more general-purpose markup language for time. It addresses the annotation of temporal expressions and events, but also the time anchoring of events (i.e. the temporal relations between events and TEs), as well as the relative ordering of events with respect to one another.

TimeML has recently been standardised to an ISO international standard for temporal information markup, ISO-TimeML (ISO-TimeML, 2007). Both the TimeML and the ISO-TimeML annotation standards define the following basic XML tags: <EVENT> for the annotation of events, <TIMEX3> for the annotation of time expressions, <SIGNAL> for capturing the textual elements that indicate a temporal relation, and the tags <TLINK>, <SLINK> and <ALINK> that capture different types of relations.

<EVENT>

The tag <EVENT> is used to mark up what was defined as an eventuality, situation or simply event in Section 2.4. It is therefore used not only for situations that happen or occur, but also for states or circumstances in which something holds true. The markable extent of an event is based on the notion of minimal chunks, therefore only one word should be annotated as the event representative. This word is chosen as being the head of the minimal chunk expressing the event, and it can be either a verb, or a noun or an adjective. The attributes of the <EVENT> tag are captured below in its BNF [1] (Backus-Naur Form).

```
attributes ::= eid eiid class pos tense aspect polarity mood
[modality] [comment]
eid ::= ID
{eid ::= EventID
EventID ::= e<integer>}
eiid ::= ID
{eiid ::= EventInstanceID
EventInstanceID ::= ei<integer>}
class ::= ' OCCURRENCE' | 'PERCEPTION' | 'REPORTING' |
'ASPECTUAL' | 'STATE' | 'I_STATE' | 'I_ACTION'
pos ::= 'ADJECTIVE' | 'NOUN' | 'VERB' | 'PREPOSITION' | 'OTHER'
tense ::= 'FUTURE' | 'PAST' | 'PRESENT' | 'IMPERFECT' | 'NONE'
aspect ::= 'PROGRESSIVE' | 'PERFECTIVE' | 'IMPERFECTIVE' |
'PERFECTIVE_PROGRESSIVE' | 'IMPERFECTIVE_PROGRESSIVE' | 'NONE'
vform ::= 'INFINITIVE' | 'GERUNDIVE' | 'PASTPART' | 'PRESPART' |
'NONE'
polarity ::= 'NEG' | 'POS' {default, if absent, is 'POS'}
```

1. The Backus-Naur Form is a formal metasyntax used to express context-free grammars (definition extracted from the Free Online Dictionary of Computing, FOLDOC, available online at http://foldoc.org/).

```
mood ::= 'SUBJUNCTIVE' | 'NONE' {default, if absent, is 'NONE'}
modality ::= CDATA
comment ::= CDATA
```

<**TIMEX3**>

The tag <TIMEX3> is used for marking up time expressions, and it received this name because it is different from both the tag <TIMEX> present in MUC and STAG, and the tag <TIMEX2> defined by TIDES. The TimeML and ISO-TimeML guidelines specify that the TIMEX3 tag should be applied to most TIMEX2 markable expressions. The main differences between the TIMEX2 and TIMEX3 markable TEs appear in the case of embedded and post-modified time expressions. Embedded TEs are no longer permitted in TimeML, and they should be annotated as two TEs connected by a signal (e.g. *<TIMEX3>three weeks</TIMEX3> <SIGNAL>after</SIGNAL> <TIMEX3>tomorrow</TIMEX3>*). Post-modified TEs should no longer be annotated so that their extent includes the post-modifying phrase or clause as in the case of TIMEX2 (e.g.*<TIMEX3>four decades</TIMEX3> of experience*).

The BNF of the TIMEX3 tag can be found below:

```
attributes ::= tid type [functionInDocument] [beginPoint]
[endPoint] [quant] [freq] [temporalFunction]
(value|valueFromFunction) [mod] [anchorTimeID]
tid ::= ID
{tid ::= TimeID
TimeID ::= t<integer>}
type ::= 'DATE' | 'TIME' | 'DURATION' | 'SET'
beginPoint ::= IDREF
{beginPoint ::= TimeID}
endPoint ::= IDREF
```

```
{endPoint ::= TimeID}
quant ::= CDATA
freq ::= CDATA
functionInDocument ::= 'CREATION_TIME' | 'EXPIRATION_TIME' |
'MODIFICATION_TIME' | 'PUBLICATION_TIME' | 'RELEASE_TIME' |
'RECEPTION_TIME' | 'NONE' {default, if absent, is 'NONE'}
temporalFunction ::= 'true' | 'false' {default, if absent, is 'false'}
{temporalFunction ::= boolean}
value ::= CDATA
{value ::= duration | dateTime | time | date | gYearMonth |
gYear | gMonthDay | gDay | gMonth}
valueFromFunction ::= IDREF
{valueFromFunction ::= TemporalFunctionID
TemporalFunctionID ::= tf<integer>}
mod ::= 'BEFORE' | 'AFTER' | 'ON_OR_BEFORE' | 'ON_OR_AFTER' |
'LESS_THAN' | 'MORE_THAN' | 'EQUAL_OR_LESS' | 'EQUAL_OR_MORE' |
'START' | 'MID' | 'END' | 'APPROX'
anchorTimeID ::= IDREF
{anchorTimeID ::= TimeID}
```

<**SIGNAL**>

The tag <SIGNAL> applies to a textual element that makes explicit the relation between two temporal entities (TE and event, event and event, or TE and TE). Signals are typically temporal prepositions like *on*, *in*, *at*, *from*, *to*, *before*, *after*, *during*; temporal conjunctions such as *when*, *while*, *before* or *after*; and special characters used in time ranges, such as - or /.

The BNF corresponding to the SIGNAL tag is:

```
attributes ::= sid
sid ::= s<integer>
```

<**TLINK**>

The tag <TLINK> represents the temporal relationship between two events, two TEs, or between an event and a time expression, and indicates how they are related in time. The temporal relations representing possible values for the attribute **relType** are inspired by Allen's set of temporal relations described in detail in Section 2.5.1.

The BNF for the TLINK tag is:

```
attributes ::= [lid] [origin] (eventInstanceID | timeID)
[signalID] (relatedToEventInstance | relatedToTime) relType
lid ::= ID
{lid ::= LinkID
LinkID ::= l<integer>}
origin ::= CDATA
eventInstanceID ::= IDREF
{eventInstanceID ::= eventInstanceID}
timeID ::= IDREF
{timeID ::= TimeID}
signalID ::= IDREF
{signalID ::= SignalID}
relatedToEventInstance ::= IDREF
{relatedToEventInstance ::= EventInstanceID}
relatedToTime ::= IDREF
{relatedToTime ::= TimeID}
relType ::= 'BEFORE' | 'AFTER' | 'INCLUDES' | 'IS_INCLUDED' |
'DURING' | 'DURING_INV' | 'SIMULTANEOUS' | 'IAFTER' | 'IBEFORE'
| 'IDENTITY' | 'BEGINS' | 'ENDS' | 'BEGUN_BY' | 'ENDED_BY'
```

<**SLINK**>

The tag <SLINK> is used for subordination relations between two events. These relations are either intensional, factive, counter-factive, evidential, negative evidential or conditional. They require deep semantic knowledge for their identification.

The BNF for the SLINK tag is:

```
attributes ::= [lid] [origin] eventInstanceID [signalID]
subordinatedEventInstance relType
lid ::= ID
{lid ::= LinkID
LinkID ::= l<integer>}
origin ::= CDATA
eventInstanceID ::= IDREF
{eventInstanceID ::= EventInstanceID}
subordinatedEventInstance ::= IDREF
{subordinatedEventInstance ::= EventInstanceID}
signalID ::= IDREF
{signalID ::= SignalID}
relType ::= 'INTENSIONAL' | 'EVIDENTIAL' | 'NEG_EVIDENTIAL' |
'FACTIVE' | 'COUNTER_FACTIVE' | 'CONDITIONAL'
```

<**ALINK**>

The tag <ALINK> is used for aspectual relations between aspectual events (e.g. *start*, *continue*) and their event arguments. The types of aspectual relations to be encoded are: initiation, culmination, termination or continuation.

The BNF for the ALINK tag is:

```
attributes ::= [lid] [origin] eventInstanceID [signalID]
```

```
relatedToEventInstance relType
lid ::= ID
{lid ::= LinkID
LinkID ::= l<integer>}
origin ::= CDATA
eventInstanceID ::= IDREF
{eventInstanceID ::= EventInstanceID}
signalID ::= IDREF
{signalID ::= SignalID}
relatedToEventInstance ::= IDREF
{relatedToEventInstance ::= EventInstanceID}
relType ::= 'INITIATES' | 'CULMINATES' | 'TERMINATES' |
'CONTINUES' | 'REINITIATES'
```

The fact that TimeML emerged as an ISO standard proves that TimeML has been widely accepted as the most important markup language for time. However, the following chapters will illustrate the fact that there is still much work needed to improve the TimeML annotation guidelines due to all the errors and inconsistencies present in the human annotation. Given its intended use in a number of applications that require access to the temporal information embedded in text, it is of utmost importance to revise the TimeML annotation scheme, with a view towards achieving better performance both in human and computer-based annotation (see Section 3.3.2).

This section has presented existing annotation schemes capturing different temporal facets of natural language texts. Among the above-described schemes, the most important and widely employed standards are TIDES TIMEX2 (Ferro et al., 2005) and TimeML (Saurí et al., 2006). The resources that have been annotated according to these standards are presented in the following section.

3.3 Annotated corpora

This section describes the existing annotated resources that are most widely employed by researchers studying different temporal phenomena. Annotated corpora are important both for linguists who want to analyse temporal phenomena, and for corpus linguists who employ the annotated data in training and evaluating algorithms for automatic temporal processing.

The corpora presented below are taken as reference by the research community, and this is one important reason for using them in the experiments described in this thesis. Their choice was also motivated by the type of annotation they contain, as typically only one corpus was developed for each annotation standard. In this way, the researchers studying methods to automatically annotate texts according to a specific standard can easily compare their results. It is however worth mentioning that there are other smaller resources available for studying different temporal-sensitive problems, but due to space restrictions they will not be presented in this work.

3.3.1 The TERN corpus

The TERN corpus (Ferro et al., 2004) is the corpus employed in the TERN 2004 competition (Ferro, 2004), whose aim was to evaluate systems capable of performing automatic TIMEX2 annotation. The TERN 2004 exercise extends the MUC definition of the TIMEX category in terms of broader coverage of expressions, and by introducing attributes that capture the meaning of a temporal expression. The corpus includes both English and Chinese data annotated according to the TIDES TIMEX2 annotation standard. The following discussion focuses on the English part of TERN, because only the English data was used

	Annotator 1	Annotator 2	Annotator 3
Partial recognition (TIMEX2)	0.973	0.972	0.915
Full extent (TEXT)	0.963	0.911	0.894
VAL	0.981	0.939	0.940
MOD	0.983	0.800	0.564
SET	0.980	0.835	0.833
ANCHOR_DIR	0.982	0.879	0.777
ANCHOR_VAL	0.942	0.856	0.728

Table 3.4: Official inter-annotator agreement figures for the TERN corpus

for experiments in this thesis.

The English TERN data were assembled from a variety of sources selected from broadcast news programs, newspapers and newswire reports, and included 767 training documents and 192 test documents. [2] It was annotated by three annotators using the Alembic Workbench (Day et al., 1997) and the Callisto annotation tool (Day et al., 2004). The entire process went through the stages of annotation, discussion and reconciliation until reaching an inter-annotator agreement of 90% or above on partially identifying TEs [3], and on the value of the VAL attribute. The inter-annotator agreement was computed by scoring each annotator against the final adjudicated gold standard generated by Lisa Ferro, the co-author of the TIMEX2 guidelines. Table 3.4 presents the scores achieved by the three annotators when their annotations were compared to the reconciled gold standard.

The scores on the row **Partial recognition (TIMEX2)** indicate the percentage of temporal expressions correctly annotated with a TIMEX2 tag by

2. These figures are extracted from an official presentation on *TERN Evaluation Task Overview and Corpus* that is available online at http://fofoca.mitre.org/tern_2004/ferro1_TERN2004_task_corpus.pdf

3. Two annotators are considered to have annotated the same TE even if their annotations match only partially. In the rest of the thesis, the numeric figures corresponding to partial matches will be attached the label TIMEX2.

each of the three annotators, in the sense that they annotated at least a part of the markable TE present in the gold standard. The task was difficult because annotators had to mark not only time nouns and numeric expressions, but also other parts of speech such as adverbs and adjectives. For this reason there was an occasional disagreement over whether something was considered markable. It was noticed that annotators missed certain TEs, particularly pre-nominals like the adjective *former* in the context *the former senator* that in many cases was not annotated as TE. However, the annotation proved to be quite accurate for time nouns and numeric expressions.

The scores on the row labelled with **Full Extent (TEXT)** indicate in how many cases the annotated span of text representing a TE is exactly the same as the extent of the TE encountered in the gold standard (the byte offsets for the start and end of the TE are the same as in the gold standard). Problems appeared because human annotators often did not look beyond the head and they did not include post-modifiers (e.g. *a year when most candidates are afraid of appearing negative*), or pre-modifiers (e.g. *almost a decade*) and determiners (e.g. *the 1960s*). They also had problems with embedding, especially in the case of appositives (e.g. *The speaker focused on* ***1955, the year he was born.***), and they were confused over where the head was in contexts like *a* ***three-hour*** *meeting.*

The following rows in Table 3.4 represent the agreement obtained when assigning values to the TIMEX2 attributes. In the case of the VAL attribute, human annotators made errors [4] when typing the value, when selecting list-items, when calculating the value of the attribute, and when using the calendar. Some

4. The source of information concerning error sources in the manual annotation process is an official presentation on *Annotating the TERN Corpus*, available online at http://fofoca.mitre.org/tern_2004/ferro2_TERN2004_annotation_sanitized.pdf

errors then propagated, as certain dates or times were saved and reused to fill in the value of VAL for other underspecified expressions. The annotators also had problems understanding the guidelines, or remembering all the details specified by the guidelines. There were cases when the annotators were not to blame for the inconsistencies in annotation, as the guidelines sometimes offer more than one choice for encoding the same thing, or the text is just too ambiguous and one has to annotate according to their interpretation (e.g. *on the night of a presidential debate*).

The MOD attribute was also subject to human error, but this was mostly because there is a low number of modified expressions in text, so the annotators were not used to specifying a value for the MOD attribute. Sometimes they did not notice that the expression was modified, or when they did notice, cases of disagreement appeared over the MOD type (e.g. for the expression *nearly 3 years* one annotator selected the MOD value APPROX and another annotator selected LESS_THAN).

The SET attribute was also subject to human forgetfulness and to disagreement over what a set expression is (e.g. set expressions were confused with generic expressions, as in *winter snowstorms*).

The annotation errors for the anchoring attributes ANCHOR_DIR and ANCHOR_VAL were due to annotators forgetting to apply them, or because they did not pay attention to all the information present in a document. There were also problems caused by making the distinction duration vs. point, or not knowing what the granularity of ANCHOR_VAL should be, so it was difficult for the annotators to be consistent especially because natural language is vague about when durations begin and when they end.

Despite all the errors that appeared during the annotation process, the TERN corpus is the most reliably annotated resource for temporal processing developed so far, and is used frequently for the evaluation of systems performing TIMEX2 annotation. No other resource bearing temporal annotation has reached the level of inter-annotator agreement achieved by the TERN data. The detailed annotation guidelines contributed greatly to this achievement.

3.3.2 The TimeBank corpus

TimeBank (Pustejovsky et al., 2006) is the human-annotated corpus marked up for temporal expressions, events, and temporal relations as a proof of concept for the TimeML standard presented briefly in Section 3.2.4. The development of TimeBank started with choosing 300 texts from a variety of media sources from the news domain including texts from the Document Understanding Conference (DUC) corpus (biographies, descriptions of single and multiple events), texts from the ACE program (transcribed broadcast news and newswire), and *Wall Street Journal* newswire texts.

These texts were initially submitted to two preprocessing stages. The first stage involved running an automatic tool for the identification of simple TEs to reduce the amount of manual labour required. The second preprocessing stage involved running a modified version of the Alembic NLP system (Day et al., 1997) to generate likely event anchors such as verb phrases, and the tense and aspect information extracted from the verb phrases.

At this point, the data were loaded into the Alembic Workbench annotation tool, and the annotators marked up existing events, temporal expressions, signals and temporal relations linking pairs of temporal entities. The TimeML information was added to the original data in the form of XML tags. The average

Tag Name	Number of Occurrences
EVENT	7935
TIMEX3	1414
SIGNAL	688
ALINK	265
SLINK	2932
TLINK	6418

Table 3.5: TimeBank 1.2 statistics for each TimeML tag

time a trained annotator spent marking up a document of 500 words was 1 hour. The annotation process was slow, lacking proper quality control like dual annotation or any attempt to achieve agreement of at least 90%, and there was no clarification or enforcement of the guidelines.

The first released version of TimeBank (version 1.1) comprises 186 annotated texts from the initial set of 300. A second version, TimeBank 1.2 was released after revising the first version, and it contains 183 documents with just over 61,000 words. It is considered to be a small corpus, in fact too small to be useful for machine learning. The statistics for each TimeML tag are found in Table 3.5.

In order to measure inter-annotator agreement, a subset of ten documents from TimeBank 1.2 was independently annotated by two experienced annotators. The agreement on tag extents was computed as the average of precision and recall with one annotator's data as the key and the other's as the response. The official figures can be found in Table 3.6.

The low inter-annotator agreement score for TLINKs is due to the large number of event-pairs that can be selected for specifying temporal links, and any two annotators working on the same text are very likely to select different pairs of entities to be linked via temporal relations. Therefore, the main problem is that annotators do not create the same TLINKS, and if they do, they only agree

Tag Name	Agreement on Tag Extent
EVENT	0.78
TIMEX3	0.83
SIGNAL	0.77
ALINK	0.81
SLINK	0.85
TLINK	0.55

Table 3.6: Inter-annotator agreement for TimeML tag extents

77% of the time on the temporal relation, as one can see in Table 3.7 detailing the inter-annotator agreement on each TimeML tag attribute both in terms of average precision and recall, as well as using the traditional Kappa statistics (Cohen, 1960).

The official inter-annotator figures reveal low agreement for certain tasks, such as deciding on the class of an event or annotating temporal relations. This illustrates the difficulty of performing TimeML annotation, but also the fact that temporal phenomena are not very well understood by humans. This lack of a clear picture over how time is expressed in text and over what is the best way to represent it formally is easily inferred from the ambiguous TimeML annotation guidelines that leave many aspects of the annotation underspecified. The existing problems concerning TimeML and TimeBank are also highlighted by other authors (Boguraev and Ando, 2005, 2006; Derczynski and Gaizauskas, 2010a), who bring additional evidence of inconsistency in the annotation of TimeBank.

3.3.3 The Aquaint corpus

The Aquaint corpus (Graff, 2002) is a new addition to the collection of TimeML-compliant corpora. This corpus is sometimes referred to as the Opinion Corpus.

Tag and Attribute	Inter-annotator Agreement	
	Precision and Recall	Kappa
EVENT.class	0.77	0.67
EVENT.pos	0.99	0.96
EVENT.tense	0.96	0.93
EVENT.aspect	1.00	1.00
EVENT.polarity	1.00	1.00
EVENT.modality	1.00	1.00
TIMEX3.type	1.00	1.00
TIMEX3.value	0.90	0.89
TIMEX3.temporalFunction	0.95	0.87
TIMEX3.mod	0.95	0.73
ALINK.relType	0.80	0.63
SLINK.relType	0.98	0.96
TLINK.relType	0.77	0.71

Table 3.7: Inter-annotator agreement for TimeML attribute values

It is very similar in content to, and uses the same specifications as, the TimeBank 1.2 corpus.

The Aquaint corpus contains 73 documents and around 38,000 words. The annotation process was similar to the one employed during the annotation of TimeBank, and probably very similar inter-annotator agreement figures apply to its development.

Future plans of the TimeML work group are to merge the TimeBank 1.2 and the Aquaint TimeML corpora, and to create a significantly larger TimeBank by using widely accepted corpus creation standards like dual annotation and revision of the annotation guidelines until reaching an inter-annotator agreement of at least 90%.

3.3.4 The TempEval corpus

The TempEval corpus (Verhagen et al., 2007), based on TimeBank 1.2, was created for the TempEval evaluation exercise organised as part of SemEval-2007 [5]. It was the first time a temporal annotation exercise was included in the SemEval/Senseval evaluation challenge.

The TempEval evaluation exercise focused on the identification of temporal relations between predefined temporal entities. The way the tasks were defined allowed a certain level of consistency in the annotation, in the sense that it was no longer left to the annotator's choice which entities should be linked with a temporal relation. The pairs of entities involved in a temporal relation were predefined, and the annotators had the simplified task of choosing the exact temporal relation between the two temporal entities. In this way, TempEval tried to overcome the main problem encountered in the annotation of TimeBank, i.e. what entities should be linked via TLINKs.

The TempEval corpus contains the same documents as TimeBank 1.2, and preserves the annotation of events and temporal expressions from TimeBank 1.2, but uses a simplified set of temporal relations, grouped according to the three separate tasks presented below:

Task A: determine the temporal relation between an event and a temporal expression situated in the same sentence;

Task B: determine the temporal relation between an event and the document creation time (DCT);

Task C: determine the temporal relation between the main events of two consecutive sentences.

5. SemEval-2007 is the fourth in the series of Senseval evaluation campaigns, aiming at evaluating the strengths and weaknesses of systems that perform different tasks related to the semantic analysis of text.

The data sets for the three tasks included the sentence boundaries, the TIMEX3 tags (including the document creation time tag), and the EVENT tags that were also present in TimeBank 1.2. The targeted set of temporal relations for each task was prepared automatically, so that all human annotators and automatic systems labelled the same TLINKs. For the first two tasks, a restricted set of events terms was used, namely those events whose stems occured twenty times or more in TimeBank.

All three tasks rely on a simplified version of the TimeML set of temporal relation labels including BEFORE, AFTER and OVERLAP. The task organisers have added to these three labels two disjunctions BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER, and an undetermined relation VAGUE. They hoped that by simplifying the labels defined in TimeML, the data preparation process would be alleviated, the complexity of the tasks would be reduced, and the inter-annotator agreement would increase. The manual annotation process was indeed about 10 times faster than for TimeBank. However, the inter-annotator agreement still remained low, being 69% for task A, 74% for task B, and 65% for task C. These figures, along with the problems identified after the competition, have convinced the organisers that the choice of relations is still problematic, and that a good direction would be to decompose each task into smaller subtasks for which detailed annotation guidelines should be defined. Section 7.3 demonstrates the feasibility of such smaller subtasks.

3.4 Approaches for TE identification and normalisation

This section describes previous approaches taken towards the identification and normalisation of temporal expressions in text. The presentation follows a chronological order and, whenever appropriate, groups similar or co-temporal approaches under a single heading.

3.4.1 Natural language interfaces for temporal databases

Ion Androutsopoulos in his work (Androutsopoulos, 1996; Androutsopoulos et al., 1998; Androutsopoulos, 2002) presents the development of a natural language interface for temporal databases (NLITDB). His NLITDB system allows users to pose temporal questions in natural language to consult an airport database. It maps English queries to a temporal extension of SQL via an intermediate semantic representation. Many temporal questions involve the use of temporal adverbials, therefore the system has the ability to identify and capture the meaning of temporal expressions to be able to generate the corresponding temporally constrained queries. The system is able to deal with punctual adverbials consisting of the preposition *at* followed by a clock-time expression (e.g. *at 5:00 pm*), with period adverbials introduced by *in*, *on*, *before* and *after*, as well as with the adverbials *today* and *yesterday*. Probably the factors that have lead to this limitation in coverage are domain specificity, high frequency of these adverbials in the controlled language used to interrogate the database, as well as the amount of work involved in describing how each case should be mapped to a semantic representation. However, Androutsopoulos's work remains among the first attempts to interpret time-related linguistic phenomena computationally.

3.4.2 Scheduling dialogues

Alexandersson et al. (1997), Busemann et al. (1997) and Wiebe et al. (1998) describe natural language processing systems that resolve temporal expressions in meeting scheduling dialogues. These systems serve as natural language front-ends for the interaction of different automated agents that negotiate and finally schedule times of appointments for their respective owners. In the context of the dialogues exchanged by the scheduling agents, many cases of underspecified or anaphoric temporal expressions are present, and these are resolved using the most recent expressions of the text already processed, and in case of failure they are resolved with respect to the time the message was sent. While Alexandersson et al. and Busemann et al. only briefly mention or describe the methods chosen for temporal resolution, Wiebe et al. performs a detailed analysis of a corpus of scheduling dialogues and develops the most appropriate focus model for temporal reference resolution. A focus model captures the most salient entities at any point in the dialogue, thus determining which previously mentioned entities are the candidate antecedents of anaphoric references. The focus model chosen as most appropriate for temporal reference resolution is recency-based. It is structured as a linear list of all times mentioned so far in the dialogue, the list being ordered by recency. Whenever an anaphoric temporal expression requires resolution, the antecedent is considered the most recent time expression in the list satisfying the constraints. The evaluation of this model on scheduling dialogues data yielded an accuracy of 81%.

3.4.3 MUC campaigns

The MUC evaluation campaigns described in Section 3.2 have contributed to driving forward research in the area of Information Extraction. The Named Entity Recognition task of the MUC campaigns included the identification of TIMEX expressions of type TIME and DATE. A few systems that have participated in the MUC NE task are briefly described below. Due to space restrictions, the focus is on MUC-7 due to its highest time-related subtask complexity and also due to the fact that the same systems participated in previous MUCs, but they kept improving in time.

The best performing MUC system comprises text handling tools developed at the Language Technology Group (LTG). The LTG MUC system (Mikheev et al., 1998) makes use of a tokeniser, part-of-speech tagger and an SGML transducer that takes certain types of SGML elements and wraps them into larger SGML elements using different resource grammars. Such a grammar is used for capturing TIMEX expressions, since these expressions are fairly structured and can be captured by means of grammar rules. The LTG system achieved the highest score of the participating NER systems: for DATE expressions the F-measure is 93.73% and for TIME expressions the F-measure is 87.07%. The authors admit to a relatively low recall for the TIMEX category due to underspecification in the guidelines and training data.

Another system that participated at MUC-7 was Facile (Black et al., 1998), a rule-based system that supports context-sensitive partial parsing and is able to categorise texts in four languages: English, German, Italian and Spanish. This system's functionality when dealing with Named Entities included tokenisation, part-of-speech tagging, database lookup and Named Entity rule application. The

novelty of this system is the new rule-based formalism defined so that the values stored in the feature vectors associated with each text token could be readily used as rule constituents. This offered better readability and allowed rules to be built using attributes arising from multiple levels of analysis. The authors report an overall F-measure of 82.25% for their NE identification approach.

A similar approach was taken by the University of Sheffield which participated in MUC-7 using LaSIE-II (Humphreys et al., 1999), a system integrated in the GATE (General Architecture for Text Engineering) platform (Cunningham et al., 1996). GATE offers a highly modular approach to language processing: for a given text it manages all the information produced by each module, provides graphical tools for visualising that information, and selecting control flow through different module combinations. For the task of NE Recognition, LaSIE-II integrates a cascade of specialised grammars that make use of part-of-speech tags and semantic tags provided by a gazetteer lookup process in order to identify a chunk of a particular category. This methodology is very similar to the finite state models advocated by most MUC participants. The result obtained by LaSIE-II for the overall NE task in terms of average precision and recall is 85.83%, and no detailed results corresponding to TIMEX entities are provided. The system developers mention that the time expression recognition task in MUC-7 was particularly hard due to the introduction of relative time expressions both for dates and times of day, saying that the task guidelines were not completely defined for this subtask.

It is worth clarifying that the MUC tasks tested the accuracy of systems in flagging time expressions, and did not require resolution of their values. The MUC tasks were also simplified by the fact that at least 30% of the dates and times in the MUC test set had a fixed format (Mani, 2003), being easy to identify

with a low number of patterns, thus justifying the choice of a rule-based approach adopted by all participating systems.

3.4.4 Mani and Wilson

In contrast to the MUC exercises, Mani and Wilson (2000) focus on resolving temporal expressions, thus bringing a novel contribution towards the normalisation of TEs. They discuss a preliminary annotation scheme, introducing the attribute VAL that would receive a value compatible with the ISO-8601 standard according to the pattern *CC:YY:MM:DD:HH:XX:SS*. In addition to the values provided by the ISO standard, they added several extensions, including a list of tokens to represent commonly occurring temporal units, such as the token *SU* for *summer*. The novel attribute VAL and the extensions defined by Mani and Wilson were later included in the TIMEX2 annotation language.

A test corpus consisting of 221 articles were hand-tagged according to this preliminary annotation scheme, with the inter-annotator agreement across 5 annotators on 193 articles being 0.79 F-measure for extent and 0.86 F-measure for assigning time values (Mani et al., 2004). Mani and Wilson's time annotation system, called TempEx, scores 0.76 F-measure in identifying time expressions. The errors are mainly caused by formats not yet implemented. When assigning values to the correctly identified TEs, the authors noted that the largest source of errors was caused by expressions that were assigned a value when they should have received none. This is called **the generic vs. specific problem** and it arises when an expression like *today* can have a specific use (meaning the day of the utterance) and a generic use (meaning *nowadays*). Generic usages should not be assigned a value, and the system automatically fills one in. To solve this problem, the authors experimented with different sets of features and a machine

learning algorithm incorporated in C4.5 (Quinlan, 1993) in order to learn rules for setting apart generic from specific usages of *today*. The best rules learned by C4.5 were then incorporated in TempEx.

Another problem identified by Mani and Wilson at the normalisation stage is **the direction problem** that concerns named expressions like *Tuesday* or *January* whose associated value should be determined according to the direction of the offset (i.e. towards the past or towards the future) from the reference time. They use the tense of a neighbouring verb to decide in which direction to look to resolve the expression.

TempEx achieves an F-measure of 0.86 at the normalisation stage on the same 193 articles inter-annotator agreement was measured on.

While the work of Mani and Wilson brings novel insights into the process of TE normalisation, it is worth mentioning that their system has its limitations both in terms of TE identification (e.g. they do not tag unanchored intervals) and in terms of TE normalisation (they only normalise date and time referring expressions, all other TE types are ignored).

3.4.5 TERN

The TERN (Time Expression Recognition and Normalisation) 2004 competition was the first exercise evaluating system performance both in terms of recognition, as well as normalisation of TIMEX2 temporal expressions using as gold standard the TERN corpus described in Section 3.3.1.

The evaluation was focused on the following three problems:

- **Detection**: refers to the ability of systems to identify at least one character belonging to a gold standard TE. This means that a system's output tag is

scored as a correct detection if it has even a minimal overlap with the tag annotated in the gold standard.

- **Bracketing (extent recognition)**: measures the ability of systems to correctly determine the full extent of a TE, for all correctly detected TEs. This means that a system's output TE must match exactly the extent of the TE annotated in the gold standard.

- **Normalisation (attribute value assignment)**: measures the ability of systems to correctly assign the correct attribute values included in the TIMEX2 tag, for all correctly detected expressions. The attributes include: VAL, MOD, SET, ANCHOR_VAL and ANCHOR_DIR, each attribute being evaluated separately.

The TERN competition was divided into two separate tasks, allowing systems to choose which problems they want to tackle:

- **Recognition only**: evaluated systems on their performance on detection and bracketing;

- **Recognition and normalisation**: involved evaluation of systems on the basis of their performance on TE detection, bracketing, and normalisation.

All systems that embarked on the recognition task approached it from a machine learning perspective.

The ATEL (Automatic Temporal Expression Labeler) system developed at the University of Colorado (Hacioglu et al., 2005) adopts a statistical approach to detect temporal expressions both in English and Chinese. Each sentence in the TERN training data is converted to a token-level representation, each token

being assigned a tag according to a bracketed representation that can incorporate embedded expressions. The possible tags are: "(*)" for a one-token TE, "O" for a non-TE token, "(*" for the beginning of a time expression, "*" for a token inside a time expression, "*)" indicates a token that ends a TE, and "((*", "*))", "((*)", "(*))" and "*)))" for different cases of embeddedness (this classification is very similar to the **BIO** - ***B****egin* ***I****nside* ***O****utside* - tagging formalism). The authors train Support Vector Machine (SVM) classifiers (Vapnik, 1995; Burges, 1998) on this converted data using several lexical (e.g. the token, its frequency in a lexicon), syntactic (e.g. phrase chunks), semantic (e.g. head words, dependency relations) and external features (e.g. the decision of a rule-based TE tagger). Then the system is evaluated on the test data, and its performance in terms of detection is 93.5% for English and 90.5% for Chinese, while the results for bracketing are 87.8% for English and 78.6% for Chinese.

A similar statistical approach modeling the TE recognition problem as a classification problem was adopted by Alias-I's LingPipe named entity annotator (Carpenter, 2004). LingPipe's entity extraction is based on a Bayesian generative model that labels each token as being the beginning of a TE, the continuation of a TE, or not included in a TE. In this model, a token/tag pair is generated probabilistically based on the previous token/tag pairs.

The best results in the TE recognition task were achieved by IBM's system (Ittycheriah et al., 2003) based on maximum entropy (Ratnaparkhi, 1999). The authors investigate learning semantic trees using a maximum entropy framework. The underlying MaxEnt semantic parser works in three stages: part-of-speech tagging, chunking and structure buiding. All the decisions in building this tree are modelled using maximum entropy models.

In contrast to the statistical approaches adopted for TE recognition, the combined task involving both recognition and normalisation was solved using knowledge-based approaches.

The most detailed system description in the published literature is that of the Chronos system developed at ITC-IRST (Negri and Marseglia, 2005). This system addresses the task with a rule-based approach, separating TE recognition (detection and bracketing) from their interpretation (normalisation). At the detection and bracketing stage a linguistic analysis (tokenisation, part-of-speech tagging, multiword recognition) of the input text, followed by rule application yield an intermediate annotation containing all the relevant information for the normalisation phase. This intermediate annotation is transformed into values for each TIMEX2 attribute during the normalisation process that relies on heuristics. With this approach, Chronos outscores all systems on several attributes (VAL: 0.87, MOD: 0.77, ANCHOR_DIR: 0.76 and ANCHOR_VAL: 0.72).

However, for detection and bracketing the best performance was achieved by AeroText (Cassel et al., 2006), a system developed by the company Lockheed Martin. AeroText recognises TEs with a hand-crafted set of rules, and then normalises them using the document creation time as the anchor for relative TEs. The normalised values are stored using an interval-based representation. The AeroText interval forms are then translated to the normalised forms required by TERN using another set of rules.

Another rule-based system that took part in TERN 2004 was the one developed at the University of Amsterdam (Ahn et al., 2005c). The authors use finite state automata for this task. The rules used for recognition are augmented with pattern matching variables to extract elements of the expression necessary to compute the normalised value, and with functions that perform

the computation with respect to the document creation time. They focus mainly on identifying the value of the TIMEX2 VAL attribute, giving only a superficial treatment to all the other attributes. Their participation in TERN made them acknowledge that a rule-based system achieves high precision, but the recall is directly correlated with the effort invested in rule development. A machine learning system can provide excellent results on the recognition task, but machine learning alone cannot solve the normalisation problem. As a result, the authors decided to optimise recognition and normalisation independently, at the same time exploring opportunities for the use of data-driven methods to solve normalisation subtasks (Ahn et al., 2005d,a). They first employ Conditional Random Fields (CRFs) (Lafferty et al., 2001) for the task of TE Identification, and then they decompose the normalisation task into five stages: lexical lookup (mapping names to numbers, units to ISO values, etc.), context-independent composition (combining the values of lexical tokens to produce a semantic representation), context-dependent classification (determining whether a TE is a point or a duration, solving the direction problem and the generic vs. specific problem), reference time tracking (finding the antecedent for anaphoric TEs), and final computation (combining the results of the previous steps to obtain a final value). The first two stages are addressed with a rule-based approach. The context-dependent classification problems are solved independently using maximum entropy classifiers (Berger et al., 1996) to decide firstly whether the TE refers to a point in time or a duration, secondly whether the TE refers to a point before, after or the same as the reference time, and thirdly whether an occurrence of *today* is generic or specific. The reference time tracking problem is resolved using two models: one that uses the document creation time as reference for all underspecified TEs, and another one that uses the most recent suitable

TE as reference time for anaphoric TEs. Again the focus is only on the value of VAL, and the results obtained using this approach are promising: the best model achieves 0.77 F-measure for VAL.

3.4.6 More recent work

An investigation of more recent work on TE identification and normalisation has revealed that the two main directions presented before have been preserved in newer work. Some authors like Mazur and Dale (2007) adopted the rule-based approach, building on previous work invested in the development of GATE (Cunningham et al., 1996), and trying to bring their own contribution at the level of TE representation. Other authors like Ahn et al. (2007) have continued improving their previous approach (Ahn et al., 2005a) by using an alternative machine learning technique: Support Vector Machines.

The most innovative recent approach applies bootstrapping to the extraction of temporal expressions from large unlabelled corpora (Poveda et al., 2009). The algorithm starts off with a set of seed examples and an unlabelled training corpus. Then it follows a repetitive cycle of extracting patterns from examples, ranking the extracted patterns, and applying the patterns to the corpus to extract new examples that are ranked and added to the initial set of seeds. The approach is novel and interesting, but unfortunately the results are much below other methods, with the bootstrapping system achieving in the best scenario 60.59%.

Most of the work described so far has focused on English, although some systems were capable of dealing with other languages: Wiebe et al. (1998) and Saquete-Boro (2005) processed Spanish texts, Alexandersson et al. (1997) and Busemann et al. (1997) dealt with German, Black et al. (1998) could

recognise TEs in German, Italian and Spanish as well as English, Negri and Marseglia (2005) were also able to annotate Italian texts with TIMEX2 tags, and Hacioglu et al. (2005) can insert TIMEX2 annotation in Chinese texts. This work acknowledges the research concerned with the multilingual dimension of TE annotation, but due to space limitations, not many systems are mentioned.

3.5 Event annotation

This section illustrates how computational research efforts have evolved in the area of event identification and annotation. The approaches presented in this section are different not only in terms of the methodology chosen to annotate events, but also from the perspective of the event definition they relied on. Early research looked at events expressed using verbs (Klavans and Chodorow, 1992) and classified them using the linguistic tests described by Dowty (1979). Later evaluation exercises such as MUC (Sundheim and Chinchor, 1993) or TDT (Allan et al., 1998) considered events to be either templates requiring their slots filled in (MUC), or instances of a topic defined by the list of stories discussing it (TDT). Most of the work that followed dealt with a more linguistically grounded notion of events that were associated with different textual extents ranging from verbs (Siegel and McKeown, 2001; Saurí et al., 2005) and sometimes nouns (Saurí et al., 2005) to clauses (Filatova and Hovy, 2001) and relationships between named entities made through a connector (Filatova and Hatzivassiloglou, 2003). All these efforts are presented in detail below.

3.5.1 Klavans and Chodorow

Klavans and Chodorow (1992) were pioneers in applying statistical corpus analysis to aspectual classification to distinguish between stative and non-stative events. They considered verbs to be the expression of events, and experimented with the 100 most frequent verbs appearing in the Brown Corpus (Francis and Kucera, 1982). Each verb was automatically assigned a numerical value representing its **degree of stativity** by using tests inspired from Dowty (1979), relying mostly on the frequency of occurrence of that particular verb with the progressive. For a given verb, its assigned value could be seen as the degree of likelihood that given any context, that verb will be used statively or non-statively.

3.5.2 MUC campaigns

In the context of the MUC campaigns, a Scenario Template task was built around extracting pre-specified event information and relating that information to the entities involved in the event. An event was seen as a template requiring its slots to be automatically filled in: an event was considered to be a relationship between participants, times, and places. A different scenario was defined in each MUC campaign, so each campaign involved developing or adapting the participating systems to a new domain. As part of the scenario definition, the participating systems received a list of events with their associated slots. Most systems relied on semantic concept hierarchies that were customised to each new domain (Humphreys et al., 1999; Yangarber and Grishman, 1999). Template slot values were filled in after the text underwent several stages of processing: morpho-syntactic analysis, translation into semantic representation, mapping the semantic information to a representation of instances, their ontological classes and

their properties according to the domain at hand, applying scenario-sensitive inference rules, and finally identifying instances that satisfied the scenario requirements and filling the slots. This highly domain-dependent manner of extracting information about events makes it difficult for researchers to achieve decent levels of accuracy: the highest score achieved for this task at MUC-7 was 50.59% (Aone et al., 1999).

3.5.3 The Topic Detection and Tracking (TDT) framework

The **TDT framework** (Allan et al., 1998) adopts a different view on events, associating an event with an instance of a topic and being defined by a list of stories that discuss it – a narrowly defined topic for search. TDT research started with a pilot study in 1996-1997, and continued with evaluations until 2004. The intentions of the TDT framework were to explore techniques that can detect the appearance of new events, and can track their reappearance and evolution. Within this framework several tasks have been defined, including a **detection task** targeting the identification of events/topics that have not been seen before, and a **tracking task** whose aim is to group together all the news stories that discuss a single event/topic.

The detection task was typically approached by reducing stories to a set of features, and for each new story its feature set is compared to those of the already seen stories. If there is sufficient difference, the story is marked as introducing a new event. One approach (Allan et al., 1999) represented each story as a vector and compared two stories using cosine similarity. The authors first experimented with agglomerative clustering by comparing each new story to

existing clusters and adding it to the most similar cluster if the similarity was higher than a threshold, or otherwise creating a new singleton cluster. Another method they used to compare a story to previously seen material was nearest-neighbour comparison. In this case incoming stories are directly compared to all the stories seen before. After locating the most similar neighbour, if the story's similarity to the neighbour exceeds a threshold, the story is declared old, otherwise it is declared a new event.

In the tracking task, participating systems were provided with a small number of stories that were known to define an event, and for each new story they were expected to decide whether the story talked about the same event or not. Jin et al. (1999) approached this task with a probability-based system that made use of three probabilistic models yielding three separate scores, all representing the probability that a new story was relevant to the topic defined through the input set of articles, then they employed logistic regression on the training data to estimate each score's weight with the aim of applying them in a linear combination of the three scores to the test data. They obtained good results, proving that such a method can reliably indicate how likely it is that a story discusses an event represented as a group of news articles.

3.5.4 Siegel and McKeown

At the same time that events were identified either using scenario templates or by grouping news articles into clusters by topic, other researchers were dealing with a more linguistically grounded notion of events. Siegel and McKeown (2001) investigate a method for automatic aspectual classification based on the assumption that a verb's aspectual category can be predicted by co-occurrence frequencies between the verb and certain linguistic modifiers. They name these

frequency measures **linguistic indicators**. A set of 14 linguistic indicators are combined for aspectual classification using three supervised machine learning methods: decision trees, genetic programming, and logistic regression. The authors experiment with two aspectual distinctions, one that classifies verbs according to stativity into states and events, and the second one that classifies events according to completedness (telicity) into culminated and nonculminated events. Separate corpora are manually annotated for each of the two classification problems. The features used for training and testing the three machine learning methods are the values corresponding to the 14 linguistic indicators calculated for each verb (except the verbs *to be* and *to have*) as being the frequency of the aspectual marker with the verb. The linguistic indicators informing the machine learning algorithms include the frequency with which verbs appear in progressive constructions, in passive constructions, in the company of *not* or *never*, modified by a temporal adverb such as *then* or *frequently*, modified by a manner adverb, modified by a duration *in*-PP, or by a duration *for*-PP. Decision trees are found to be the most successful method for the two types of aspectual classification targeted in this work, achieving an accuracy of 93.9% for the state vs. event classification, and 74.0% for the culminated vs. nonculminated event classification.

3.5.5 Filatova

Filatova and Hovy (2001) resolve the problem of event identification by breaking sentences into event-clauses, as they consider the clause to be the expression of an event. This simple approach works reasonably well, and is employed in a larger system dealing with time-stamping events that will be presented in more detail in Section 3.6.

Another variant of the notion of event is experimented with by Filatova and Hatzivassiloglou (2003), an event being seen as a relationship between participants, times and places, i.e. a connection between two named entities made through a connector. Following an empirical study aimed at providing an operational definition of events, the authors use the observations to develop an algorithm for detecting, extracting and labelling events. As part of their study of event annotation, they asked students to mark text passages that describe events in news articles without providing them with any definition of an event. Substantial disagreement was observed as to what should be marked as an event. In terms of extent, 62% of the annotated text spans represented a sentence [6], 24% a clause, 14% multiple sentences. By examining the annotations, the authors decided to choose the sentence as the scope of an event and to anchor events on named entities representing participants, locations or times. The algorithm starts with identifying named entities and selecting only the sentences that included more than two NEs. They extract all possible pairs of NEs and the in-between words are examined in order to preserve only the verbs and the nouns that are hyponyms of *event* or *activity* in WordNet, thus ensuring a high probability that the pair of NEs together with its connector represent an event. The connector list is filtered so that only highly frequent connectors are kept, and the NE pairs that are not connected by frequent connectors are eliminated. A graph of connections is built and then undergoes a merging process that groups together edges representing pairs of NEs that share a common endpoint and substantially similar connectors. The results indicate that this is a promising approach for obtaining a shallow interpretation of event participants and their relationships.

6. Sometimes a short prepositional phrase was not included in the annotated extent, but the authors do not provide examples of such cases for a better understanding.

However, a drawback of this method is that it does not locate events when unnamed participants are mentioned.

3.5.6 TimeML-motivated research

The development of TimeML and TimeBank have stimulated work in the area of TimeML-compliant event identification and annotation. EVITA (Saurí et al., 2005) was the first system capable of annotating events according to the TimeML annotation standard. The functionality of EVITA breaks down into two steps: event identification and event annotation. The event identification part targets verbs, nouns and adjectives. The verbs labelled as events by EVITA are all non-auxiliary verbs present in some lexical inventories, except the verb *to be* and generic verb usages (e.g. verbs that appear with bare plural subjects). Events expressed by nouns are identified by lexical lookup (each noun is checked if it is present in any of the 25 subtrees selected from WordNet as including nominal events), followed by a disambiguation phase in the case of nouns that appear in WordNet as both an event and a non-event (the disambiguation is based on rules learned by a Bayesian classifier trained on SemCor). In what adjectives are concerned, only those that are annotated as events in TimeBank are marked as events by EVITA. The next part involves the annotation of the previously identified events with the TimeML EVENT tag and its corresponding attributes. The values of the attributes *pos*, *tense*, *aspect*, *vform*, *polarity* and *modality* are directly derivable using linguistic rules and pattern matching from the morpho-syntactic information provided by a POS-tagger and a syntactic parser. The attribute *class* is assigned the class that was most frequently associated with that particular event in TimeBank. The evaluation of EVITA was carried out on TimeBank, yielding 80.12% for event identification, 89.95% for the *pos* attribute,

92.05% for *tense*, 97.87% for *aspect*, 98.26% for *polarity*, 97.02% for *modality*, and 86.26% for *class*. The result for the task of event identification (80.12%) was compared by the authors to the inter-annotator agreement achieved by graduate students for the task of annotating verbal events (80%) and nominal events (64%). EVITA's results demonstrate very good performance, but since they were produced on the same corpus that EVITA was trained on (TimeBank), it is highly likely that they overestimate EVITA's performance when confronted with new texts.

In an attempt to generalise the TimeML annotation strategy and to overcome the small size of TimeBank and its inappropriateness for machine learning approaches, Boguraev and Ando (2004) exploit un-annotated corpora using a word profiling technique for the tasks of event identification and class assignment. Word profiling collects and compresses co-occurrence frequencies of words and features which capture the typical neighbours that a word has – both in terms of distance and syntactic relations. These word profiles are then used as features by a classifier – Robust Risk Minimisation (Zhang et al., 2002) – that solves the problem of event identification in a similar manner to named entity recognition - it decides if a word is: inside an event-chunk, the last word of an event-chunk, or outside any event (this is similar to the **BIO** tagging formalism). The classification task (i.e. identifying the value of the *class* attribute corresponding to an event) is solved by training the classifier to distinguish between a higher number of labels: for each event class to be distinguished there are two labels corresponding to the word being inside or at the end of a chunk belonging to that particular event class, to these adding the label for words situated outside any target chunks. However, despite the fact that certain features used by the classifier are derived from un-annotated data, the classifier is trained and tested

on TimeBank. The evaluation results show 80.3% F-measure for identifying events, a performance similar to EVITA's, and 64.0% for assigning them a class.

A similar approach is taken by Bethard and Martin (2006), Bethard (2007) and March and Baldwin (2008) who also formulate the event identification and classification tasks as machine learning problems. Bethard and Martin (2006) and Bethard (2007) use Support Vector Machines to assign to each word in a document a **BIO** label indicating whether the word is inside or outside an event, labels which are augmented with event class information (for example, in the case of REPORTING events, the labels would be B_REPORTING and I_REPORTING). The set of features includes: the targeted word, affix features, morpho-syntactic features including dependency features, negation, temporal features, WordNet hypernym features, as well as features that capture co-occurrence statistics of the verbs and their direct objects. At the evaluation stage, the authors implement two baselines, one that simply assigns to each word the label with which it appears most frequently in TimeBank, and a simulation of the EVITA system that no longer allows EVITA to use the same data for training and testing. The results for event and class identification measured on 18 documents extracted from TimeBank are 50.2% for the first baseline, 50.9% for the simulation of EVITA, and 57.9% for the system of Bethard (2007). March and Baldwin (2008) deals only with event identification by using features that are more superficial from a semantic perspective, such as word and POS context and order, word grouping, stop words, NE information and achieves an F-score of 76.4% for event identification.

Llorens et al. (2010b) analyse the contribution of semantic role labelling to TimeML event recognition and classification. They employ as a learning method Conditional Random Fields (Lafferty et al., 2001), and their features consist of

morpho-syntactic features, WordNet-based features and semantic role features that include a word's semantic role, governing verb, role-verb relation and role configuration. A semantic role labeler was applied to TimeBank to facilitate the extraction of values for the semantic role features. This classifier achieves for recognition an F-measure of 81.40%, and for classification 64.20%. The authors also evaluated the classifier without the semantic role features and noticed a decrease of 2.73% in the overall performance, concluding that semantic roles are useful in the identification and classification of events.

This section has presented different research directions adopted for the task of event identification. Each approach was modelled in accordance to the event definition it dealed with. Despite considering various structures as events, when looking at events from a linguistic perspective most researchers concentrated on verbs or their dominance domain – the clause, and even the sentence – as the typical expression of an event. The study conducted by Filatova and Hatzivassiloglou (2003) confirmed that this point of view is universally valid for humans. In their study, 86% of the text spans annotated by students who were asked to mark text passages that describe events without being given an event definition represent a clause or a sentence. One can therefore conclude that verbs and their dominance domain, which can be clauses or sentences, are central to the study of events. This is an important reason why the study of events pursued as part of this research and presented at large in Chapter 6 focuses on verbs.

Having covered previous work concerned with the annotation of temporal expressions and events, the next section looks at how temporal relations among these entities have been addressed in the past.

3.6 Temporal relation identification

The most difficult challenge for researchers working in the area of temporal annotation is finding methodologies for annotating the relations between time expressions and events or between events and other events. Research in this area has matured from the initial approaches to time stamp events and to annotate corpora using various representations of temporal relations to more complex approaches targeting the automatic identification of temporal relations. All these efforts will be described in detail in the following.

3.6.1 Time stamping events

Approaches for time stamping events typically attempt to associate a calendar time or time interval with some or all events present in a text.

MUC evaluation campaigns, in addition to the task of identifying TEs of type TIME and DATE (see Section 3.4.3 for more details), also required the assignment of a TIME or DATE expression to the slot LAUNCH_DATE of the predefined scenario template corresponding to MUC-7 rocket launch events. Since this was only a secondary task, not much detail is provided by the authors of participating systems as to how it was addressed. In the case of the LaSIE-II system (Humphreys et al., 1999), the authors approached the Scenario Template task by using rules organised in cascade grammars to build a discourse model, these domain-specific rules guiding the extraction of the values for the required slots. All systems achieved quite low results on this particular slot, reflecting the difficulty of the task.

As part of the MUC task, times were only assigned to certain scenario events. A more general approach attempted to assign a time point or time interval to

absolutely every event in the text. Filatova and Hovy (2001) experimented with both manually and automatically decomposing news stories into their constituent event clauses and assigning time stamps to each event clause following an analysis of the temporal adverbials and of the verbal tense information characterising each event clause. A time-stamp assignment was considered to be correct whenever the event clause's real time-stamp was included in the time interval provided by the system for that event clause. The time-stamper was evaluated both on manually annotated event clauses achieving 77.85% accuracy on a set of 158 clauses, and on correctly identified clauses extracted from the output of the syntactic parser: in this case the accuracy was 82.29% on 96 clauses.

Schilder and Habel (2001) made the transition between time-stamping events and proper temporal relation identification by defining a set of temporal relations and assigning them to time-event pairs. Their approach relied on the assumption that relations between times and events should be marked only when they are explicitly signalled by prepositions, or whenever they are syntactically implicit. The authors designed a temporal annotation system for German that assigned a default temporal relation to each pair of event – TE connected via a preposition, while the inclusion relation was assumed for cases when no preposition was present. Their system marking only temporal relations between events and temporal expressions involved in a direct syntactic relation was evaluated on a small corpus of 10 German news articles and achieved an overall precision and recall of 84.49%.

The ultimate goal of event time-stamping approaches was to anchor all events in a text on a time line. However, natural language texts rarely specify the exact position an event should have on a time line, and mostly provide a partial ordering between events. This calls for a different representation that does not

require a total event ordering, and that does not leave out temporal relations that are explicit in text. Efforts to define such representations and to demonstrate their suitability by applying them to natural language texts are presented in the following section.

3.6.2 Annotation of corpora with temporal relations

Katz and Arosio (2001) propose a semantic formalism suitable for the annotation of intra-sentential temporal relations, and then apply it to syntactically annotated sentences with the aim of creating a treebank annotated with temporal relations and morpho-syntactic information. This resource could then prove useful for examining the influence of lexical and syntactic structure on temporal ordering. The manual annotation process targets pairs consisting of a verb and the speech time, as well as pairs of verbs expressing states or events that are situated in the same sentence. Each verb is associated with a temporal interval and the relations among these intervals are reduced to either precedence or inclusion. The authors reported an inter-annotator agreement of 70% on a set of 50 sentences.

Setzer and Gaizauskas (2002) promote a novel temporal representation for a text that takes the form of a time-event graph, the nodes of the graph being either times or events and the arcs representing event-event and time-event temporal relations. The authors argue for the superiority of this representation over the "time-stamping" paradigm that associates a time with an event. They annotate a trial corpus of 6 newswire articles using the STAG annotation scheme that adheres to the time-event graph representation (for more details see Section 3.2.3). The annotation takes place in two stages. The first stage covers the annotation of events, time expressions, signals and temporal relations that are either explicitly expressed or syntactically implicit. The second stage relies on the information

annotated at the first stage and automatically derives all possible inferences, thus enriching the annotation with new temporal relations. At this stage, the human annotator is only prompted to manually specify a temporal relation when the system is unable to infer a relation for a given pair of events or events and times. The process then continues cyclically until every event-event and event-time pair in the text are temporally related. This annotation experiment has shown that the task is very difficult for human annotators, and that the low inter-annotator agreement is due to several causes: imprecision/incompleteness of the guidelines, imperfect annotator understanding of the task, difficulty of establishing a temporal relation in some cases, annotator fatigue, and annotator carelessness.

The efforts of Setzer and Gaizauskas to define an annotation scheme have proved essential for the development of the generally adopted standard for temporal annotation TimeML, and of the TimeML proof of concept: the TimeBank corpus. Both TimeML and TimeBank have been presented in detail in Sections 3.2.4 and 3.3.2, respectively. In the following the focus will be on approaches dealing with the automatic identification of temporal relations.

3.6.3 Automatically identifying temporal relations

Mani et al. (2003) propose a machine learning-based approach for temporally anchoring and ordering events in news. Events are associated with clauses, and the authors use several heuristics to associate with each clause a reference time value **tval**, the concept of reference time being the one proposed by Reichenbach (1947). Then they train a statistical classifier to order the event denoted by a clause with respect to the **tval** associated with that clause (equivalent to classifying the temporal relation between the event/clause and the **tval** into one

of the following classes: AT, BEF, AFT, or undefined). Based on the predictions made by the classifier, the authors infer a partial ordering of the events situated in the same document. This approach achieves 59% accuracy in assigning a **tval** to a clause, 84.6% accuracy in finding the temporal relation between an event and its associated **tval**, and 75.4% F-measure in partially ordering events whenever the temporal relations between them and their associated **tvals**, as well as the temporal order of the two **tvals** allowed a relation to be inferred.

In a later publication (Mani and Shiffman, 2005), the authors describe their efforts of simplifying the task of manually and automatically annotating temporal relations by focusing on ordering pairs of consecutive clauses where either both clauses are in Past Tense, or the first clause is in Past Perfect and the second one in Past Tense. In their experiment, 8 subjects were presented with pairs of clauses exemplifying the two tense sequences and they were asked to specify the order of the two central clause events by selecting one of the following six relations: *Entirely Before*, *Entirely After*, *Upto*, *Since*, *Equal*, *Unclear*. The inter-annotator kappa agreement observed for this annotation task was 0.5, the conclusion drawn by the authors being that such fine-grained distinctions are hard for people to make. The authors then further simplified the task by collapsing the categories *Entirely Before* and *Upto* into *BEF*, and *Entirely After* and *Since* into *AFT*, observing an increase in inter-annotator agreement to a kappa of 0.61. The annotated data was then used as training and test data for a classifier that provided an accuracy of 58.07% in ordering two successive Past Tense clauses using the coarse-grained set of relations, and an accuracy of 70.38% in finding the temporal relation between a Past Perfect and a Past Tense clause.

Following these efforts to simplify the task of temporal relation identification, Mani and his collaborators (Mani et al., 2006, 2007) take a different approach

by first trying to overcome the data sparseness problem, and then training a classifier to determine the temporal relation between two temporal entities. The authors merge the two corpora bearing TimeML annotation – TimeBank and the Aquaint Opinion Corpus – into a corpus they call OTC, and then they expand the set of temporal relations annotated in OTC by using a temporal closure component SputLink (Verhagen, 2004) to derive new implied temporal relations from the ones already annotated. This closure component increases the number of temporal relations by more than 11 times when compared to the original human annotation present in OTC. A maximum entropy classifier is then trained both on the data before closure, as well as on the closed data, the results showing that the closure caused an increase of 15% in the accuracy to identify event-event temporal relations. However, following a careful examination of the data, the authors realised that following the closure process, duplicate vectors were generated and included in the training/test data, and also there were certain overlapping feature values between the training and test data due to shared context. After addressing these issues, the accuracy of the classifier trained on the closed data dropped both for event-event and event-time temporal relations by approximately 8% and 10%, respectively, below the accuracy of the classifier trained on the unclosed data. The results emphasised a need for thorough analysis of the effect of closure on data used for training and testing classifiers for temporal relation identification.

Lapata and Lascarides (2004) associate the task of identifying sentence-internal temporal relations among clause pairs with the task of identifying the marker that has the highest probability of linking the two clauses. The authors extract from the BLLIP corpus (Charniak et al., 2000) all main-subordinate clause pairs where the main clause is linked to the subordinate clause with one of the following temporal markers: *after*, *before*, *while*, *when*, *as*, *once*, *until*, *since*.

They then train and test several machine learning models using features extracted from the main and subordinate clauses to be able to predict the marker that connects the two clauses, their accuracy being of 70.7% on this task. It should be noted that with this approach one is not able to directly obtain the temporal relation between the two clauses, as the markers predicted by the models are often ambiguous (see Section 7.2 for details on how the present work deals with ambiguous temporal markers to aid the temporal relation identification process).

In their later work, Lapata and Lascarides (2006) use the models derived from their previous work on selecting the appropriate marker for a pair of clauses to predict a reduced set of TimeML temporal relations comprising *BEFORE*, *INCLUDES*, *ENDS*, *BEGINS*, *SIMULTANEOUS*. They assigned a temporal relation to each clause pair extracted from BLLIP as described above. This assignment was unique for unambiguous markers (e.g. for *once* clauses, the relation was BEGINS), and randomly generated for ambiguous markers (e.g. for *when* clauses, a random choice was made between BEFORE, INCLUDES and SIMULTANEOUS, while maintaining the proportion equally among the three labels). A classifier was then trained on this data and tested on TimeBank, reaching an overall f-measure of 45.8%. These results show that one can infer temporal information from corpora that is not semantically annotated in any way.

Chambers et al. (2007) describes a two-stage machine learning architecture for the identification of event-event temporal relations. The first stage deals with identifying the event attributes tense, aspect, modality, polarity and event class by training a Naive Bayes classifier. The event attributes obtained at the first stage are then used together with other features in a second stage to classify the temporal relation between two events with an SVM classifier (Chang and Lin,

2001). The novelty of this work is the use at the second stage of imperfect feature values such as the ones obtained during the first stage, which still produces a small improvement over the results of other authors that used human-annotated feature values (Chambers's method achieves 65.48% for event-event temporal relation identification, while Mani's method using perfect feature values achieves 62.5% on the same OTC data set).

Other authors like Boguraev and Ando (2005); Vasilakopoulos and Black (2005); Li et al. (2004), have also explored the use of machine learning for temporal relation identification. Most of the work dedicated to automatic temporal relation identification has been stimulated by the two evaluation exercises TempEval and TempEval-2 organised as part of the SemEval 2007 and SemEval 2010 evaluation exercises on semantic evaluation.

TempEval

TempEval (Verhagen et al., 2007) was the first evaluation exercise that focused on temporal relation identification. It was organised in the context of SemEval 2007, a wider semantic evaluation campaign whose aim was to establish a benchmark for various semantic tasks of interest at that point in time. The TempEval exercise consisted of three tasks that tested the capability of participating systems to relate an event and a TE located in the same sentence, an event and the TE representing the Document Creation Time (DCT), and two events located in consecutive sentences. The data used for this exercise consisted of a simplified version of TimeBank, in the sense that only certain events and event attributes were preserved, and a simplified set of temporal relations was used (consisting of: *BEFORE*, *AFTER*, *OVERLAP*, *BEFORE-OR-OVERLAP*, *OVERLAP-OR-AFTER*, and *VAGUE*). The test data included the events and

temporal expressions together with their TimeML annotations, as well as pairs of temporal entities for which the temporal relation was supposed to be identified automatically. Six participating systems have approached the three TempEval tasks using different methods.

Four of the participating systems adopted a machine learning approach for solving the three tasks (Hepple et al., 2007; Min et al., 2007; Cheng et al., 2007; Bethard and Martin, 2007), with the most popular classifier being Support Vector Machines (Hepple et al., 2007; Min et al., 2007; Cheng et al., 2007; Bethard and Martin, 2007). However, the feature engineering process employed in these approaches involved rules of varying complexities to derive values for their syntactic and semantic features not explicitly annotated in the data.

The other two participating systems took a rule-based approach, by relying on a deep syntactic analysis of the texts. XRCE-T, the system developed at XEROX (Hagege and Tannier, 2007), relied on a syntactic analyser that was extended to deal with temporal expressions and with associating TEs with the events they modify just as thematic roles are attached to predicates. These associations were then used to order events in certain syntactic configurations. A more complex approach was adopted by the author of the present work in the system that achieved the best results at TempEval (Puşcaşu, 2007b). The approach is described in detail in Chapter 7.

The lessons learned from TempEval were that some of the tasks were not well defined (Verhagen et al., 2007, 2009) and they proved difficult to carry out, as illustrated by the relatively low inter-annotator agreement, and that the disjunctive and VAGUE labels should have not been included in the target set of temporal relations (Lee and Katz, 2009). The TempEval organisers considered that the definition of the first task should be changed from linking all events

in a sentence with all TEs situated in the same sentence to more syntactically motivated subtasks that would link temporal entities according to syntactic considerations, such as syntactic dominance, argument structure and discourse structure. The results of the TempEval competition were also analysed by Lee and Katz (2009) who concluded that only three labels should be used as target temporal relations: BEFORE, AFTER and OVERLAP. Some of these issues were addressed in TempEval-2.

TempEval-2

TempEval-2 (Verhagen et al., 2010) was organised in the context of SemEval 2010 and consisted of six tasks. Unlike the first TempEval, the tasks now targeted not only temporal relations, but also the identification of TIMEX3 time expressions and of TimeML events. Four of the six tasks involved determining the temporal relations holding between an event and a TE syntactically dominated by the event, between an event and the DCT, between two main events in consecutive sentences, and between two events involved in a syntactic dependency relation. Eight teams participated in this competition, but only three teams attempted all tasks, and it is surprising that their results did not demonstrate improvement over the results obtained in the first TempEval, despite the fact that certain tasks were simplified.

UzZaman and Allen (2010) identify temporal relations using a Markov Logic Network classifier and linguistically motivated features generated by their rule-based system for finding TEs and events in text. They participated with two systems and their systems achieved the best results for two temporal relation tasks.

Another top performer who obtained the best result in the event-DCT

temporal relation task was TIPSem (Llorens et al., 2010a), a system that employs Conditional Random Fields (Lafferty et al., 2001) as a machine learning technique for categorising temporal relations. The authors argue that the use of semantic roles among their features improved the capability of learned models to generalise rules.

The system that achieved the best scores in predicting temporal relations between main events and syntactically dominated events was NCSU (Ha et al., 2010), a Markov Logic-based system that used besides the typical lexico-syntactic features, a set of features capturing lexical relations between words extracted from VerbOcean (Chklovski and Pantel, 2004) and WordNet (Fellbaum, 1998).

To sum up, all the systems participating in TempEval-2 adopted a machine learning approach, either using Markov Logic (UzZaman and Allen, 2010; Ha et al., 2010), Conditional Random Fields (Llorens et al., 2010a; Kolya et al., 2010), or Maximum Entropy classifiers (Derczynski and Gaizauskas, 2010b). Adding more semantic-based features had a beneficial impact on system performance, but still the results were not encouraging given that the tasks were simplified when compared to the first TempEval.

3.7 Conclusions

This chapter described different approaches adopted by researchers in their attempts to solve the main problems involved in temporal processing: temporal expressions, events and temporal relations. Their work relies on several annotation schemes that were presented in detail in Section 3.2, and on resources annotated according to these standards captured in Section 3.3.

The main approaches taken towards the identification and normalisation of temporal expressions were described in Section 3.4. The computational treatment of events was presented in Section 3.5, while Section 3.6 focused on how temporal relations have been addressed in the literature.

Chapter 4

Temporal Expression Identification

4.1 Overview

Chapter 2 presented from a theoretical perspective the three main types of temporal entities that should be involved in any attempt to capture the temporal dimension of natural language texts. Chapter 3 provided an overview of the existing resources and computational approaches for the identification of these three types of temporal entities in news articles. This chapter together with Chapter 5 describe the methodology adopted in this research to solve the problem of **temporal expression (TE) annotation**.

The automatic TE annotation process involves two processing stages. The first stage is concerned with identifying the textual extent of the temporal expressions present in the processed text, and is normally referred to as **temporal expression identification**. The second stage of the annotation process is called temporal expression normalisation, and its aim is to find the value that the expression designates or is intended to designate. This chapter describes the approach taken in the present work towards temporal expression identification,

while the focus of Chapter 5 is the normalisation process.

The current chapter starts with a classification of the temporal expressions this research deals with. Section 4.2 describes in detail the most common types of TEs one can encounter in natural language texts. Having familiarised the reader with the targeted entities, the remainder of this chapter and Chapter 5 continue with a description of the automatic annotation process.

The methodology adopted in the TE identification process is detailed in Section 4.3. Section 4.4 compares the results obtained by using different knowledge sources in the TE identification algorithm. The capabilities of the TE identifier are then extended so that it can also perform TIMEX3 annotation according to the TimeML guidelines (Saurí et al., 2006), and the changes involved in this process are described in Section 4.5. The chapter finishes with conclusions.

4.2 Classification of temporal expressions

A deep understanding of the types of temporal expressions that can be encountered in a natural language text is required to be able to develop a computer system that can approximate what a human does towards the interpretation of expressions that refer to time. To this end, different sets of annotation guidelines were outlined for creating normalised representations of temporal expressions in text. This research relies mainly on the specifications encountered in the widely employed scheme for temporal annotation TIMEX2.

The TIMEX2 annotation scheme, as well as other schemes for TE annotation, distinguishes between expressions capturing when something happened (**position in time**), how long something lasted (**duration**), or how often something occurs (**frequency**). Another important distinction is made between expressions that

can be normalised relying only on themselves alone (these expressions are known as **fully specified**, **context-independent** or **absolute times**), and expressions that require the value of another TE serving as an anchor for determining which particular time is meant (these expressions are known as **underspecified**, **context-dependent**, or **relative TEs**). An example of a fully specified TE is the expression *twelve o'clock January 5, 2008* ([4.1]) that embeds all the information necessary for its normalisation. In contrast, there are underspecified expressions, such as *the following day*, ([4.2]) that need another fully specified TE to help anchor them on a timeline. If the two examples below appear as two consecutive sentences in a text, then the fully specified TE *twelve o'clock January 5, 2008* would be the anchor for the expression *the following day* and would help in determining the calendar point corresponding to the underspecified TE, i.e. *January, 6, 2008*.

[4.1] *John returned to work* ***twelve o'clock January 5, 2008***.

[4.2] *Mary started work* ***the following day***.

The classification of temporal expressions outlined in the present work relies on the theoretical distinction between the three main TE classes presented in Section 2.3. According to this distinction, TEs can indicate position in time (in the following this class of TEs will be called **CALPOINT**), duration (**DURATION**) or frequency (**FREQUENCY**). Apart from these three TE classes, the TIMEX2 annotation guidelines (Ferro et al., 2005) also mention expressions that refer in general terms to the past, present and future ([4.3]), as well as expressions that do not indicate a specific time ([4.4]).

[4.3] *The* ***present*** *problems do not allow progress.*

[4.4] *It has been* ***a long time*** *since they have seen each other.*

In this research, the TEs generically referring to the past, present and

future are grouped in the class called **TOKEN** (this name was chosen because these expressions are normalised using a pre-defined token). The term **UNANCHORABLE** will be used for those expressions that, according to the guidelines, are to be annotated, even if they do not indicate a specific time. These denominations were introduced because there is no consensus in the literature.

The TE classification presented below is one of the original contributions of this research as it synthesises the criteria widely used in the literature to indicate different processing methods that are applicable to various types of TEs, but which have never been put together in a comprehensive classification of TEs.

This research distinguishes the following classes of temporal expressions:

4.2.1 Calendar points (CALPOINT)

These expressions indicate the temporal location on the timeline, and they are also known as points in time or calendar points. They can be specified up to a certain level of detail, known as the **precision** or the **granularity** of the TE. This work will mostly use the term granularity to refer to how precise a TE is (i.e. the temporal expression *Monday* is at day level, while the expressions *2008* or *four years ago* are at year-level). Certain character codes are employed for specifying the granularity of a temporal expression: more details on these codes can be found in Table 4.1.

The calendar points can be further classified into:

Fully specified TEs

The fully specified temporal expressions can be classified according to their granularity:

Level of precision	Granularity code
millennium	ML
century	CE
decade	DE
year	Y
month	M
week	W
day	D
hour	H
minute	MIN
second	S

Table 4.1: The character codes corresponding to the granularity of a TE

- **Millennium level** (example [4.5])

  A TE fully specified at millennium level can be expressed using ordinal numbers and the lexical trigger *millenium* (*the 2nd millennium*, *the first millennium*).

  [4.5] ***The 2nd millennium*** *was a period of time that commenced on January 1, 1001, and ended on December 31, 2000.*

- **Century level** (example [4.6])

  A TE fully specified at century level can be expressed using ordinal numbers and the lexical trigger *century* (*the 17th century*, *the twentieth century*).

  [4.6] *During* ***the 17th century*** *the population of England and Wales grew steadily.*

- **Decade level** (example [4.7])

  The same decade-level TE can be expressed in different ways, for example the meaning of *the 1960s* is also captured by *1960's*, *60s*, *60's*, *Sixties*, *the sixties*.

[4.7] *Dancing deteriorated in **the 1960s** into group chaos.*

- **Year level** (example [4.8])

  Temporal expressions fully specified at the year level can be expressed either using the year alone (*1998*, *'98*), or preceded by a determiner and the word *year* (*the year 2008*). The context might determine what type of year value the TE should be annotated with. Besides the simple year type that is most commonly encountered (e.g. *1999*), one might come across financial years (e.g. *financial year 1998-1999*), copyright years (e.g. *c.1998, copyright 1998*), or years before the start of this epoch (e.g. *700 A.D.*, *18 BC*).

  [4.8] *They collaborated closely in **2008**.*

- **Month level** (example [4.9])

  At month level, fully specified TEs are normally expressed using either numeric patterns (*12/2008*) or using the full or abbreviated month names (*January 2008*, *Sept 2007*). Also part of this class are the TEs mentioning a year division either taking the form of a season (*winter 2008*), quarter (*the third quarter of 1999*) or a year-half (*the first half of 1998*).

  [4.9] *They returned to the UK in **January 2008**.*

- **Week level** (example [4.10])

  Week level TEs are quite rarely encountered in text in their fully specified form, but whenever they appear they include a week number accompanied by the word *week*, together with the month and the year, or only the year that particular week is part of.

  [4.10] *This Easter falls in **week 17, 2009**.*

- **Day level** (example [4.11])

  There are a wide variety of forms for a day level fully specified TE. Various numeric patterns are often encountered for this type of TEs (*24.2.97*, *24.02.97*, *2.24.97*, *02.24.97*, *02.24.1997*), and in many cases finding out the exact date from such a numeric expression is mainly a localisation problem (e.g. the expression 02/03/2010 can refer to 3rd of February 2010 or to 2nd of March 2010 depending on whether it is encountered in an American or British English text.). TEs fully specified at day level can also be expressed using either cardinal or ordinal numbers for the day slot, the full or abbreviated month name and a numeric value for the year slot (*May 19th, 2008*, *1st of September 2008*). This type of expressions can be either preceded or followed by the full or abbreviated day-of-week name corresponding to that particular date (*Monday, May 19th, 2008*).

  [4.11] *The accident took place on* ***the 2nd of December 2008****.*

- **Hour level** (example [4.12])

  Various patterns fully specified up to the day level can be combined with different ways of expressing the time (e.g. *12 o'clock*, *six o'clock in the evening*, *2 pm*, *14 hours*, *6 a.m.*, etc.) to obtain a TE fully specified at the hour level.

  [4.12] *The meeting is at* ***1 o'clock January 21, 2009****.*

- **Minute level** (example [4.13])

  The minute slot is typically combined with the various ways of expressing the hour (e.g. *18:30*, *6:30 a.m.*, *half past six in the evening*, etc.), and the resulting sub-expression together with a sub-expression fully specified at day level yield a fully specified expression at minute level (e.g. *half past six in the evening,*

Monday, May 19th, 2008).

[4.13] *Another meeting is scheduled at* ***13:30, 19/11/2008****.*

- **Second level** (example [4.14])

  From time to time TEs are specified up to the second level. Typically this is encountered in time-stamps formed using numeric patterns (as in [4.14]), but they are not restricted to fully numeric patterns (e.g. *19:28:20 11 Dec 98*).

  [4.14] *The engines stopped exactly at* ***16:01:57, 12/31/1998****.*

The temporal expressions indicating the time of the day (i.e. having the granularity hour, minute or second) can include reference to a timezone (e.g. *1618 GMT 11 Apr 99*, *11-15-98 1305EST*, *12/31/1998 16:01:57.14 Eastern Time*, etc).

Deictic TEs

This class of TEs includes expressions that refer to particular times relative to the Speech Time Point (see Section 2.5.4 for more details). Deictic TEs are underspecified in that they require a fully specified TE to provide the reference value with respect to which their final value is computed by using a function that is self-contained in the deictic TE itself. The Speech Time serves as anchor for expressions belonging to this class. In the case of newswire texts, the Speech Time is considered to be the Document Creation Time (DCT), and only in sentences that contain time-stamped reported speech is the Speech Time overridden by the time of the reporting event. In example [4.15] extracted from an article dated *Friday, 9th of October 1998*, the Speech Time initially set as being the DCT should be overridden by the time the reporting event takes place (i.e. *Thursday*, whose normalised value should be *Thursday, 8th of October 1998*),

Function	Usage	Description
add	**add(teValue, noOfUnits)**	This function adds the number of units **noOfUnits** to the normalised value of a TE (**teValue**), the result being a normalised TE value that is temporally located after **teValue** at a distance of **noOfUnits**. Both arguments of the **add** function should refer to the same granularity, in the sense that if **teValue** has for example a day level granularity, then **noOfUnits** should refer to the number of days to be added to **teValue**. For example, given a **teValue** of **"2009-02-24"** and a **noOfUnits** of **"3"**, the result of **add("2009-02-24", "3")** would be **"2009-02-27"**.
subtract	**subtract(teValue, noOfUnits)**	This function subtracts the number of units **noOfUnits** from the normalised value of a TE (**teValue**). For example, given a **teValue** of **"2009-02-24"** and a **noOfUnits** of **"3"**, the result of **subtract("2009-02-24", "3")** would be **"2009-02-21"**.
coerceTo	**coerceTo(teValue, granularity)**	This function constrains the normalised value of a TE (**teValue**) down to a desired granularity. For example, given a **teValue** of **"2009-02-24"** and a **granularity** of **"M"** standing for month, the result of **coerceTo("2009-02-24", "M")** would be **"2009-02"**.
getSubunit	**getSubunit(teValue, indexSubunit)** or **getSubunit(teValue, namedSubunit)**	The **getSubunit** function sees the value associated to a TE as a composite unit formed by joining together a set of subunits of higher granularity, and extracts from this set the subunit identified either by its index in the set or by its denomination. For example a TE at year granularity with the **teValue** of **"2009"** can be seen as the set of months indexed from **"1"** to **"12"**, and in this case the function **getSubunit("2009", "2")** points to the second month of 2009 - **"February 2009"** - normalised as **"2009-02"**. The same year can be seen as the set of seasons including **"SP"** (spring), **"SU"** (summer), **"FA"** (fall) and **"WI"** (winter), and in this case the function **getSubunit("2009", "SU")** yields the **"summer of 2009"** normalised as **"2009-SU"**. Therefore the **getSubunit** function locates in a given cycle the element identified either by its index in the cycle or by its name.

Table 4.2: Temporal functions used for the interpretation of underspecified TEs

and the deictic expression *yesterday* should be normalised with respect to the time of the reporting event, and should therefore receive the value *Wednesday, 7th of October 1998*.

[4.15] *"The rebels entered Iranian territory* ***yesterday****", Deputy Premier Bulent Ecevit told reporters on* ***Thursday****.*

Each deictic TE embeds a function that requires an external argument – the anchor (for deictic expressions this is the Speech Time) – for finding its actual value. This function results from the composition of the following functions: **add**, **subtract**, **coerceTo** and **getSubunit**. More details on these functions can be found in Table 4.2.

The expressions in the class DEICTIC can be further classified into:

- **Deictic adverbials**

  These include the temporal adverbials *today*, *yesterday* and *tomorrow* and they

require as anchor an expression that is fully specified at the day level or higher (i.e. hour, minute or second level). The temporal functions [1] embedded by these adverbials are as follows: for *today* the function is **coerceTo(anchor, "D")** (extracting from the anchor the day-level value), for *yesterday* the function is **subtract(coerceTo(anchor, "D"), "1")** (the value should be obtained by deducting from the anchor one day), and for *tomorrow* the function is **add(coerceTo(anchor, "D"), "1")** (the value should be obtained by adding one day to the anchor).

- ***THIS* + temporal trigger**

  The deictic expressions formed using *this* and a temporal unit (i.e. *millennium*, *century*, *decade*, *year*, *month*, *week*, *day*, *hour*, *minute*) embed the function **coerceTo(anchor, granularity)**, where the second argument is the granularity of the temporal unit (e.g. for *this week*, the function guiding its normalisation is **coerceTo(anchor, "W")**, indicating that the value to be extracted from the anchor TE should be at week level). When *this* appears with a temporal proper name referring to a day of the week (e.g. *Monday*) or to a month (e.g. *April*), the function guiding the normalisation is **getSubunit(coerceTo(anchor, "W"), dowIndex)** for days of the week (dowIndex is the index of the named day within the week, e.g. *1* for *Monday*), and **getSubunit(coerceTo(anchor, "Y"), monthIndex)** for months (monthIndex is the index of the named month within the year. e.g. *4* for *April*). In a similar way expressions involving *this* and a season (*this winter*), quarter (*this quarter*), or day-part (*this evening*) embed functions of the form **getSubunit(coerceTo(anchor, granularity), namedSubunit)**,

1. In the rest of the chapter, the notation of functions will use lowercase names for variables (placeholders) and quotation marks to designate actual values.

where **granularity** is the neighbouring granularity at a lower level of precision than the expression itself, and **namedSubunit** is the value associated with the subunit named in the expression (i.e. for *this evening*, the function would be **getSubunit(coerceTo(anchor, "D"), "EV")**). The main idea is that one first needs to extract from the anchor the temporal cycle that includes the subunit in question, and then to attach the subunit to the identified cycle to obtain the final normalised value of this type of deictic expressions.

- ***LAST/PAST* + temporal trigger**

  Expressions of this subclass are formed with *last* or *past* and a temporal unit, temporal proper name or a temporal noun, in a similar manner to the previous subclass. The only difference to the previous subclass is that the function denoted by this type of expressions is **subtract(coerceTo(anchor, granularity), "1")**, where **granularity** is determined as described above (e.g. for *last week*, the function would be **subtract(coerceTo(anchor, "W"), "1")**).

- ***NEXT/COMING* + temporal trigger**

  The way expressions of this type are formed and assigned a function is similar to the previous subclass, with the difference that they are formed by joining *next* or *coming* with a temporal unit, and that the function is this time **add(coerceTo(anchor, granularity), "1")** (e.g. for *next week*, the function would be **add(coerceTo(anchor, "W"), "1")**).

- **Quantified temporal units + *AGO***

  By combining a quantified temporal unit, or just simply a plural temporal unit, with *ago*, the resulted expressions are deictic and should also be anchored to the Speech Time. The function assigned to the TE in this case would be

subtract(coerceTo(anchor, granularity), noOfUnits), where **noOfUnits** is the number of units that should be deducted from the anchor TE (e.g. for *three months ago*, the function would be **subtract(coerceTo(anchor,"M"), "3")**). In the case of generic quantifiers (e.g. *several years ago*) or bare plurals (e.g. *years ago*), the value of **noOfUnits** would be the placeholder *"X"*, illustrating its lack of specificity (e.g. **subtract(coerceTo(anchor, "Y"), "X")**).

Dependent TEs

This class includes expressions whose values are also reliant on other TEs, but differ from those in the previous class through the fact that they are dependent on the discourse context and their anchor TE is the nearest Reference Time introduced in the discourse. Considering the example [4.16] and supposing it is extracted from an article dated *Monday, 5th of October 1998*, the expression *the following day* belongs to the class DEPENDENT and its anchor is the nearest Reference Time, i.e. *Thursday*, and therefore should receive the value *Friday, 2nd of October 1998*.

[4.16] *John went to Germany on* ***Thursday****, and came back* ***the following day****.*

The class DEPENDENT includes the following subclasses:

- ***THE/THAT*** **+ temporal trigger**

  The dependent expressions formed using the determiners *the* or *that* followed by a temporal unit (e.g. *year*, *week*, etc.) embed the function **coerceTo(anchor, granularity)**. TEs including *the* or *that* followed by a temporal proper name (e.g. *Tuesday*, *March*, etc.), or by a temporal noun (e.g. *summer*, *weekend*, *morning*, etc.) embed the function **getSubunit(coerceTo(anchor,**

granularity), namedSubUnit), where **granularity** and **namedSubUnit** are obtained as described in the case of the DEICTIC TEs. The only difference to the TEs in the class DEICTIC is that the dependent expressions are anchored to the Reference Time, and not to the Speech Time.

- ***PREVIOUS* + temporal trigger**

  These expressions are similar to the deictic ones introduced by *last* or *past*, the difference being made by the time they are anchored to (e.g. *the previous week* should be anchored to the most prominent time at that point in the discourse, while *last week* is always anchored to the Speech Time).

- ***FOLLOWING* + temporal trigger**

  They are also similar to the deictic ones introduced by *next* or *coming* and differing only in the *anchor* time (e.g. *the following month*).

- **Quantified temporal units + *BEFORE/EARLIER***

  These TEs manifest the same behaviour encountered in the case of the deictic expressions ending in *AGO*, but in this case the anchor is provided by the Reference Time (e.g. *two weeks before*).

- **Quantified temporal units + *AFTER/LATER***

  The expression resulted by combining a quantified (or just plural) temporal unit with either *after* or *later* embeds the function **add(coerceTo(anchor, granularity), noOfUnits)**, the arguments being the anchoring time, the granularity of the TE, and the number of units that should be added to the anchor TE (e.g. in the case of the expression *five weeks later*, the function would be **add(coerceTo(anchor, "W"), "5")**). For generic quantifiers and bare plurals **noOfUnits** would be *"X"*.

- ***THEN***

  Certain occurrences of *then* should be annotated and should receive a context-dependent value from the nearest Reference Time Point. Those occurrences of *then* that should be annotated and normalised are called anaphoric. Such an instance of *then* is present in example [4.17], and its anchor and value received during normalisation is given by the expression *January 2008*.

  [4.17] *John met Mary in* ***January 2008****, and since* ***then*** *he has not seen her again.*

TEs flexible in terms of anchoring (FLEX_ANCHOR)

The temporal expressions in this class are partly specified, in the sense that they lack certain parts of their value to gain the status of fully specified. They are flexible in what anchoring is concerned, as they can either be anchored to the Speech Time or to the nearest Reference Time depending on the context. Any TE not included in the previous subclasses that lacks at least the value of the year slot can be considered as belonging to this class (e.g. *Monday, May 19th*; *September 11*; *23/12*; *April*; *winter*; *the fourth quarter*; *Monday*; *late on Wednesday*; *midnight*; *a cold winter evening*; *afternoon of September 11*; *late Thursday night*; *ten minutes to four*; *half past five*; *6:30 a.m.*; *5 o'clock EST Friday afternoon*; *3:45 and 30 seconds*)

Embedded TEs (EMBEDDED)

This class contains those expressions that embed the extent of another TE functioning as their anchor ([4.18]). In such cases both expressions should be annotated, with the anchoring expression being contained within the extent of the complete phrase. The anchoring phrase can belong to any of the previously

described subclasses, and its normalisation should be done according to the rules governing its subclass. The normalisation of the complete phrase can only take place after the anchoring phrase is normalised. The value of the complete phrase is computed in relation to the value of the anchoring phrase.

[4.18] *Mary will leave* <***ten days from*** <***today***>>.

4.2.2 Duration denoting temporal expressions (DURATION)

An expression of duration indicates a period of time, providing information on how long something lasted. Durations that refer to specific periods of time can be oriented or anchored with respect to certain points in time. A expression denoting duration can be of the following types:

Simple durations

This subclass contains typical expressions denoting duration (e.g. *five weeks*), formed by adjoining a quantifier (e.g. *several*, *three*, *many*) and a temporal unit (e.g. *year*, *week*, *hour*). They can also comprise besides a quantifier and the temporal unit, the word *long*. Certain expressions included in this class can be formed recursively by allowing two or more typical durations to be conjoined using coordination (e.g. *three weeks and two days*).

Age denoting TEs

Expressions denoting age are normally formed by combining a simple duration with the word *old*, or by using a possessive pronoun together with abbreviated

forms typically used for expressing decades (e.g. *her 80s*, *their 50's*). One can also encounter cases that employ the construction *the age of* in expressing ages (e.g. *the age of 25*).

Anniversaries

Anniversaries are normally days when certain events are celebrated. This research places them under the DURATION heading due to their resemblance to AGE expressions, and also due to the fact that, if they were included in the CALPOINT class and given the value of a calendar point, the information capturing the offset of the calendar point from the initial event would be lost. By annotating an anniversary as a duration, one can recover using the values of the attributes the date when the celebrated event happened, as well as when the celebration takes place. By considering the expression *the 50th anniversary of their wedding* in example [4.19] as representing a duration, one can easily infer that the wedding took place in June 1959, and that in June 2009 there is a celebration of 50 years from the event.

[4.19] *In June 2009, they will celebrate* ***the 50th anniversary of their wedding***.

Deictic durations

Deictic durations are expressions indicating a period of time that should be anchored with respect to the Speech Time. They include expressions formed using *last*, *past*, *next* or *coming* in conjunction with quantified temporal units (or bare plurals) (e.g. *the last three years* in [4.20]). Their value is given by the expression itself, and the calendar point they are anchored to is given by the Speech Time.

[4.20] *Mary had no holiday in* ***the last three years****.*

Dependent durations

Dependent durations are expressions indicating a period of time that should be anchored with respect to the nearest Reference Time introduced in the discourse. This subclass includes expressions formed by adjoining the words *previous* or *following* with quantified temporal units (or bare plurals) (e.g. *the following weeks* in [4.21]).

[4.21] *During* ***the following weeks*** *he recovered from the operation.*

4.2.3 Frequency denoting temporal expressions (FREQUENCY)

Expressions conveying frequency capture how often something occurs. They can be expressed either using frequency adjectives or adverbs (e.g. *annually* as in [4.22]), or using *every/each* in conjunction with a temporal unit (e.g. *every month*, *each hour*). Plurals of temporal proper names (e.g. *Septembers*, *Saturdays*) can also express time frequency. Expressions of type frequency require no anchoring in time.

[4.22] *John goes* ***annually*** *on a fishing trip.*

4.2.4 Generic references to past, present or future (TOKEN)

This class comprises generic expressions referring to past, present or future (e.g. *previously*, *the present time*, *the future* as in [4.23]). The name of the class comes from the fact that these expressions receive as value a token indicating whether

the expression refers to the past (*PAST_REF*), present (*PRESENT_REF*), or future (*FUTURE_REF*).

[4.23] *Nobody knows what* ***the future*** *might bring.*

4.2.5 Unanchorable temporal expressions (UNANCHORABLE)

These expressions should not receive any value during the annotation process, as they are normally ambiguous in terms of the precise time they refer to. They include the following subclasses:

Holidays (HOLIDAY)

Expressions of type HOLIDAY refer to names of festivals, holidays and other occasions that have a name recognised in a certain community (e.g. *Thanksgiving*, *Diwali*, *Christmas*). Such expressions should be marked in text, but normally they should not be assigned a value, unless explicitly provided in the context. One can argue that HOLIDAY-type expressions often have an associated value (e.g. *Christmas* is always on the *25th of December*), but the TIMEX2 annotation guidelines indicate that such expressions should be assigned a value "only when that value can be inferred from the context of the text, rather than from cultural and world knowledge" (Ferro et al., 2005).

Fuzzy expressions without a precise value (UNSPECIFIED)

Certain expressions are too fuzzy to receive a value, even if they do possess a temporal flavour ([4.24]).

[4.24] *No demonstrations were allowed during* ***the election period****.*

Event-anchored temporal expressions (EVENT_ANCHORED)

An event-anchored time expression is an expression that requires knowledge about the time of an event in order for its value to be fully specified ([4.25]). These expressions normally receive no value, unless the time the embedded event took place is very obvious in the immediate context.

[4.25] *The firefighters returned home* ***three days after the fire****.*

After seeing what classes of temporal expressions one can come across in natural language texts, the following section focuses on the method used in this research to identify automatically all these types of expressions.

4.3 Methodology for the identification of temporal expressions

Automatic identification of temporal expressions is an Information Extraction task, and more specifically a Named Entity Recognition subtask, whose goal is to automatically extract chunks of text that carry direct or inferred temporal information. The simplest way to approach this task is by targeting only simple date and time values that typically adhere to a small number of patterns used when expressing time. But this task, even if it can superficially seem simple, involves the recognition of a wide variety of TEs, and this makes the task much more interesting and challenging.

As in any other Information Extraction task, two approach types can be distinguished for the identification of temporal expressions: rule-based and data-driven (discussed in detail in Section 3.4). Rule-based methods rely on handcrafted rules resulting from extensive data analysis, while data-driven

approaches employ machine learning either for sequence labeling using BIO-tagging, or to classify a certain syntactic constituent (e.g. noun phrase) as belonging to the class of TEs or not. For the TE identification task only, both techniques can be successfully employed as long as sufficient training data is available, and each technique has advantages and disadvantages. On the one hand, rule-based systems can yield very high precision, but significant human effort invested in rule development is required to achieve good recall. On the other hand, machine learning methods can provide very good results if a large enough labelled corpus is available.

When it comes to TE normalisation, rule-based approaches are by far more appropriate than data-driven techniques, for several reasons. First, there are a potentially unlimited number of temporal values that can be associated with the identified TEs during the normalisation process. It is very unlikely to be able to train a classifier that could correctly guess the values to be assigned to the identified TEs. Then, a significant number of TEs require non-local context for their normalisation (this is the case of deictic and dependent TEs). Even more problematic is the fact that a large number of TEs need to be associated with a temporal function that takes as argument the TE serving as anchor, and then they require significant temporal computation that accounts for contextual information. A machine learning approach would find it difficult to make the connection between form and content by using both context and world knowledge. However, one can successfully employ machine learning for solving small subtasks in the process of TE normalisation (Ahn et al., 2005a), but these can only render a good performance when included in a rule-based framework.

Since the goal of the present work is to develop a system that performs both identification and normalisation of TEs, a rule-based approach was adopted. The

approach taken in this work is to separate the two main stages in the annotation of TEs, identification and normalisation. Even if separated, the two stages are not independent of one another, as the information gathered at the first stage is essential for the second stage. At the identification stage, not only is the extent of a TE identified, but also all the pieces of information made explicit in the expression itself are extracted to be used in the normalisation process either for inclusion in the final normalised value of the TE, or for computing that value by considering both context and world knowledge.

In the following, the modules involved at the identification stage are presented.

4.3.1 Rule-based identification of TEs

Finite state automata have been successfully employed in many tasks that involve partial parsing or chunking (Abney, 1996). The task of TE recognition can be viewed as a partial parsing task, and it can be tackled with good results by using rules that simulate the functionality of a finite state automaton (Negri and Marseglia, 2005; Ahn et al., 2005c). As previously stated, a rule-based approach was also adopted in the present work to address the task of TE identification.

Unlike other rule-based systems that start with a linguistic pre-processing of the input text (Negri and Marseglia, 2005; Ahn et al., 2005c), the first phase of this TE identifier involves applying patterns to raw text which did not undergo any pre-processing. Ahn et al. (2005c) identify problems if the part-of-speech tagging and syntactic chunking are performed before running the identification rules due to tokenisation issues (especially in the case of punctuation signs) that prevent certain TE identification patterns from matching. To avoid such problems, the choice made in this research was to first apply the identification patterns, and then to check the syntactic correctness of the identified TEs.

The rule development process was guided by the set of markable temporal expressions defined by the TIMEX2 guidelines (see Section 3.2.2), covering the types of expressions described in Section 4.2. As TEs are normally signalled by certain lexical triggers that appear in the input text, a lexicon including these triggers was built, with every trigger being assigned a class name and an associated value. For example, the proper noun *July* is considered to be a lexical trigger belonging to the class *MONTH*, and associated with the value *7* (*July* is the *7*-th month of the year). Certain configurations of numeric expressions (e.g. 13/08/2004) that carry a meaning related to time could also be considered lexical triggers, but they are not included in the lexicon. A small subset of lexical triggers was presented in Section 2.3.1.

The lexicon contains 842 entries corresponding to temporal units (e.g. *day*), months of the year (e.g. *September*), days of the week (e.g. *Tuesday*), seasons (e.g. *winter*), names of decades (e.g. *fifties*), expressions used generically to refer to past, present or future (e.g. *nowadays*), modifiers (e.g. *more than*), generic quantifiers (e.g. *many*), determiners (e.g. *the*), ways to express parts of units (e.g. *half*), ordinal numbers (e.g. *first*), numbers expressed in words (e.g. *sixteen*), and other words and expressions that typically appear within TEs. More than half of the lexicon covers time zones (e.g. *GMT*, *Western Standard Time*) and names of holidays and special days (e.g. *Christmas*, *Semana Santa*).

The classes of triggers from the lexicon are employed in writing regular expressions of high complexity capable to recognise a wide variety of temporal expressions. Approximately 250 complex rules have been defined. These rules not only identify sequences of words representing a TE, but they also generate semantic representations for each TE at the time of matching. The semantic representations take the form of typed feature structures that depend on the

semantic class of the TE, with features such as the temporal unit and value for durations, or the year, month, day of the month for calendar points that specify explicitly these values. In the case of deictic and dependent TEs, the semantic representation includes the function to be used in computing their final value. This representation can cover most expressions and is able to cope with phenomena such as under-specification.

The values associated with the lexical triggers in the lexicon, as well as those embedded in the surface form of the expression itself (in most cases numeric), together with the pattern matched, all contribute to the semantic representation of a TE.

The rule development process was guided by a philosophy of only adding rules which were (nearly) certain never to generate errors. This could be characterised as a high precision, and possibly lower recall, approach to the creation of TE identification patterns. As already stated, the rule development process started with defining rules to cover all the cases described in the TIMEX2 annotation guidelines. Afterwards the patterns were applied to a set of news articles, and a series of iterations followed involving error analysis and rule refinement in an attempt to provide a reasonable level of generalisation and to avoid introducing errors.

The developed set of regular expressions proved extremely powerful, but it was rather easy to notice their limitations when it came to natural language and its open-ended nature. The TEs identified in the pattern-matching process can be seen as chunks forming a basic level of constituency. However, to identify the correct extent of TEs, access to a higher level of constituency is needed, and this can only be achieved through a syntactic analysis of the TE's surrounding context.

The following section focuses on a module that, by gaining access to morpho-syntactic information, is able to bring improvements to the TE identification process.

4.3.2 Checking syntactic correctness

The second stage of the system is concerned with generating syntactically valid TEs. Considering that the full extent of a TE should be a well-formed syntactic constituent, there is a need for a module that checks the syntactic well-formedness of the entities identified at the previous stage. This module, apart from checking the syntactic correctness of the identified TEs, should also modify their extent so that they adhere to the TIMEX2 specifications defining the correct extent of a TE.

As already mentioned in Section 2.3.1, the full extent of a TE should either be a noun, adjective, adverb or any of the corresponding phrases (noun, adjectival or adverbial phrases). TEs cannot be prepositional phrases or clauses, so they cannot start with a preposition or a subordinating conjunction (e.g. *after Friday*, *before they meet on Monday* are disallowed as temporal expressions). Premodifiers of temporal expressions such as determiners (e.g. *a great day*), and postmodifiers such as prepositional phrases or subordinate clauses should be included in the time expression (e.g. *the year of the elections*, *the year when he started University*). The appositives that may appear after a TE are not to be included in the expressions tag, but, if they contain trigger words, they are to be tagged separately. In the case of temporal range expressions (*from 1990 to 1999*), and conjunctions (*today and tomorrow morning*) or disjunctions (*six months or a year from now*) of time expressions, the points should be tagged separately, even if they share modifiers. In other cases more than one lexical trigger can appear

within the same TE, and in such contexts where more indicators are present, the number and full extent of the corresponding TEs are determined using the following rules defined in the TIMEX2 annotation guidelines (Ferro et al., 2005):

- one TE is created if there are no intervening words between the temporal terms that qualify a unit of time (e.g. *<twelve o'clock midnight>*), if the terms are connected with the preposition *of* (e.g. *<the evening of December, 31>*) or if the prepositions *to*, *till*, *after*, *in* are used for expressing a certain point of time in a day. In these cases, but also in the case of the "MONTH DAY, YEAR" format, the expression containing all the terms should be tagged as a single unit.

- multiple TEs with embedding appear in two cases. One is when the larger TE denotes an offset to another TE included in it. In this case two tags are created with the one corresponding to the anchoring phrase contained within the extent of the tag of the complete phrase (e.g. *<two weeks from <next Tuesday>>*). The second case is characterized by the larger TE being a possessive construction. If both the possessive phrase and the phrase that it modifies are time-denoting expressions, then two tags are created, and the possessive phrase tag is contained within the extent of the complete phrase tag (e.g. *<<This year>'s spring>*).

- multiple TEs without embedding are created in cases other than those described above, meaning that temporal phrases appearing in close proximity (like appositive phrases, range expressions, and conjoined expressions) are tagged as independent phrases. Although tagged independently in terms of the extent, there is a dependency in terms of the value. The expression with finer granularity inherits the value of the coarser-grained expression. This inheritance

happens regardless of the relative ordering of the two expressions (e.g. *<8.00 pm> on <Friday>*).

According to these TIMEX2 specifications, the functionality of the module that checks the syntactic correctness of the TEs identified at the previous stage is as follows. Firstly, the input text is parsed using Connexor's FDG parser (Tapanainen and Jarvinen, 1997). This parser returns information on a word's part of speech, morphological lemma and its functional dependencies on surrounding words, and this syntactic information is used by the system with the assumption that it is 100% correct. However the evaluation and error analysis presented in Section 4.4 show that this process introduces errors as well. Secondly, errors introduced by the rule-based TE identification module are corrected by using syntactic information. Such errors include:

- TEs starting with a determiner that is syntactically dependent on a noun that follows the TE. In these cases the determiner should be removed from the TE (e.g. the rule-based TE identification module provides as output for the noun phrase *the night shift* the TE *the night*, but syntactic information indicates that *the* is actually linked to the noun *shift* rather than *night*, and as a consequence the determiner is eliminated from the TE).

- verbs wrongly annotated as TEs due to being homographs with certain lexical triggers (e.g. the verb *present* could be mistaken due to the same spelling for the noun or adjective *present* referring to the present time). These cases are removed from the set of TEs previously identified.

- TEs that can be extended to their left with pre-modifiers that syntactically depend on any word included in the TE (e.g. the TE *night* is initially annotated

in the sentence *It was a long night*, but after considering syntactic information it is extended to the entire NP headed by the trigger word, yielding the expression *a long night*).

- TEs that can be extended to their right with post-modifiers such as prepositional phrases or relative clauses syntactically dependent on the head of the expression (e.g. the TE *an evening* is initially annotated in the sentence *It was an evening he will never forget*, but syntactic information leads to the inclusion of the relative clause in the extent of the final TE *an evening he will never forget*).

- embedded TEs that are identified by the rule-based TE-identifier either as two separate TEs that should be annotated as one TE embedding another TE (e.g. <*two weeks from* <*next Tuesday*>>), or detects only the larger TE, without annotating the embedded one (<<*this year*>*'s spring*>).

This section has focused on using syntactic information in order to check and correct the extent of the TEs identified at the pattern-matching stage. However, some problems of a semantic nature cannot be solved either using patterns, or syntactic information. This is the case of the adverb *then*, capable of manifesting several semantic values, of which only the anaphoric one should be labelled as a time expression. A novel methodology developed as part of this research to disambiguate each usage of *then*, and only annotate the anaphoric cases, is presented in the following section.

4.3.3 Disambiguation of *then*

Overview of the problem

The adverb *then* is among the most frequent English temporal adverbs, and it has great communicative strength, easily expressing one or another semantic category (or more than one simultaneously). It can play the role of a linking adverbial, but also realise the semantic role of time. At the temporal expression identification stage, it is important to separate the anaphoric usages of *then* from the non-anaphoric ones, as only anaphoric *then* should be annotated as a TE. Little previous work has tackled the automatic identification and temporal resolution of anaphoric *then*, being merely looked at from a linguistic perspective (Schiffrin, 1990; Thompson, 2005).

As part of the present effort directed towards better TE identification, an empirical investigation of all possible usages of *then* was conducted (Puşcaşu and Mitkov, 2006). The individual study of *then* in the context of TE identification/normalisation can be likened to the individual study of *it* in the anaphora resolution process (Evans, 2000). The adverb *then* can either refer to a time given in the context (synonym with *at that time* – anaphoric usage, [4.26]), or, quite commonly, mark the next event in a sequence ([4.27]), denote a result/inference ([4.28]) or mark enumerations ([4.29]), as well as antithesis ([4.30]). Only the first usage of *then* should be annotated as a TE and receive a temporal value, but the second use is also important for the task of temporally ordering events. The accurate recognition of a particular usage of *then* thus contributes to all fields in which temporal information is a concern, whether it be event-based information organization, text summarisation or question answering.

[4.26] *New Delhi exploded a nuclear device in 1974, but has not undertaken*

any nuclear tests since ***then****.*

[4.27] *The state has to hold 51 percent of Lietuvos Nafta for three years but can* ***then*** *bring its share down to 34 percent.*

[4.28] *"One of the great lessons of history is that if America is prepared to fight many wars and greater wars and any wars that come,* ***then*** *we will fight fewer wars and lesser wars and perhaps no wars at all", said Dole.*

[4.29] *He has the opportunity, the motivation, and* ***then*** *the courage to do it.*

[4.30] *You promise to help me,* ***then*** *you let me down!*

Corpus Annotation

Following the theoretical investigation described above, five categories of uses of *then* are distinguished in this research: ANAPHORIC, TIME_REL, INFERENTIAL, ENUMERATIVE and ANTITHETIC. These classes are used to annotate a corpus of 1,000 newspaper articles randomly extracted from the Reuters Corpus (Rose et al., 2002), with the word *then* appearing at least once within each document. The annotated data contains 410,391 words and 1,173 occurrences of *then*. This corpus has been annotated by two annotators to measure the interannotator agreement, thus gaining an insight into the complexity of the problem and the validity of the designed categories. To facilitate the markup of the usage type of *then*, only paragraphs containing the word together with one preceding paragraph (extracted to provide context) have been presented to the annotators. Each human annotator has been asked for a decision regarding the class *then* belongs to. The annotators had to decide among six classes: ANAPHORIC, TIME_REL, INFERENTIAL, ENUMERATIVE, ANTITHETIC and ERROR. The class ERROR has been introduced as cases have been observed during annotation where *then* was incorrectly used instead

of *than* due to typing errors. The kappa agreement observed between the two annotators is 0.86. Since the annotators have never agreed on antithetic usages of *then*, this has led to the conclusion that the antithetic value always overlaps with other semantic values, being difficult to set apart. It has also been observed that the capacity of *then* to express more semantic categories simultaneously accounts for many differences of opinions between the two annotators.

Considering that the main aim of this investigation was to identify only anaphoric usages of *then*, inter-annotator agreement has also been measured when distinguishing only between two types of usages: ANAPHORIC and NON-ANAPHORIC. The kappa agreement between the two annotators is in this case 0.92.

A machine learning approach for the disambiguation of *then*

The machine learning approach presented below was employed first for distinguishing among the six classes initially annotated, and then for setting apart anaphoric from non-anaphoric usages of *then*. For the purposes of the work described here, the implementation of k-nearest neighbours included in the software package called TiMBL (Daelemans et al., 2004) was used for experiments. The features used for training the classifier were defined so that their values could automatically be extracted from any text syntactically parsed (in this case with Connexor's FDG parser). These features are: the relative position of *then* with respect to the closest subject and predicate, the parts of speech of the two preceding and one following words, the part of speech of the word *then* is syntactically dependent on, the tenses and distances measured in number of words to the preceding and following verb phrases, collocational features, and a feature capturing whether or not *then* is possibly included

within a noun phrase. Evaluation using the leave-one-out approach on the data that included the cases agreed on by both annotators, accounting for 1,070 occurrences of *then*, revealed an accuracy of 87.75% for distinguishing among the six classes, and 91.58% for the coarser-grained classification between ANAPHORIC and NON_ANAPHORIC usages of *then*. This binary classifier significantly outperforms the baseline that considers each occurrence of *then* as belonging to the majority class NON_ANAPHORIC, and makes a correct class assignment in 71.30% of the cases.

When applying this classifier trained on the cases agreed on by both annotators to the occurrences of *then* encountered in the TERN training data, an accuracy of 85.00% is achieved when trying to distinguish between the six annotated classes. The binary classifier that distinguishes only between ANAPHORIC and NON_ANAPHORIC usages of *then* achieves on the TERN training data an accuracy of 86.25%.

An empirical approach for the disambiguation of *then*

After gaining a better linguistic insight into the issue of *then*, this work proposes a new empirical method that achieves better results when disambiguating between ANAPHORIC and NON_ANAPHORIC *then*. Previous linguistic investigations of *then* (Thompson, 2005) accounted for the interaction between the syntax and semantics of *then*. The author provides a natural explanation for how the position of *then* affects its temporal interpretation. This explanation relies on the syntax of tense and, more specifically, on the relationship between the meaning and phrase structure of tense. A Reichenbachian approach to the semantic representation of tense is assumed, where tenses are composed of three times: the Event Time, the Speech Time and the Reference Time (see Section 2.5.4). Reference Time is

represented as a semantic feature associated with the head of the Aspect Phrase (AspP). Event Time is a semantic feature associated with the head of the Verb Phrase (VP), and Speech Time a feature associated with the head of the Tense Phrase (TP).

Thompson has shown that *then* has different readings (co-temporal or ordered) depending on whether it is adjoined to the VP (i.e. the Event Time) or to the AspP (i.e. the Reference Time). Whenever *then* is in clause-final position, it is adjoined to the VP and has a co-temporal interpretation (corresponding to the ANAPHORIC usage). In clause-medial position, *then* is adjoined to the AspP, and the same happens when it appears in initial position (in this position it is considered to be fronted from medial position). The author shows that clause-initial and clause-medial *then* modify the Reference Time and induce an ordered interpretation (NON_ANAPHORIC).

On the basis of this linguistic analysis, an empirical rule-based disambiguation algorithm for *then* was designed and implemented by the author of this thesis as an alternative to the machine learning method described above. This algorithm tries to guess the semantics of *then* from its syntax:

- if *then* depends syntactically on a preposition (with which it forms a PP), the preposition requires *then* to be temporally anchored, thus ANAPHORIC (examples of prepositions: *since*, *from*, *until*, etc.)

- if *then* is dominated syntactically by a noun, it is ANAPHORIC (it is either included in an NP - *the then president* - or it is a temporal adjunct in a relative clause that depends on an NP - *the person who was then ruling the country...* - or is part of a reduced relative or appositional construction - *Mr. X, then president of the US, decided to enforce this law.*)

- any other occurrence of *then* in clause-final position is ANAPHORIC

- all other occurrences of *then* in clause-initial or clause-medial position are NON_ANAPHORIC.

This algorithm was implemented and tested on the cases of the training data agreed by both annotators (i.e. 1070 occurrences of *then*), and the correct distinction between the ANAPHORIC and NON_ANAPHORIC usage of *then* was made in 1023 cases, yielding an accuracy of 95.60%. It was also evaluated on the TERN data, with 73 correctly identified cases out of 80 occurrences of *then*, thus an accuracy of 91.25%. A detailed error analysis can be found in the following section (i.e. Section 4.4).

This section presented a highly detailed investigation of the adverb *then*, in an attempt to identify its anaphoric usages, and annotate them accordingly with TIMEX2 information. For the purpose of this research, effort was invested in developing a corpus of usages of *then*, with a kappa inter-annotator agreement of 0.92 measured when two annotators looked at only two usages of *then*: ANAPHORIC vs. NON_ANAPHORIC. This corpus was then employed in a machine learning experiment, by training a classifier to distinguish between ANAPHORIC and NON_ANAPHORIC usages. As a result of deeper linguistic investigations of *then*, another method for the disambiguation of *then* emerged, this time knowledge-based, and its results show that one can reliably distinguish anaphoric usages with an accuracy of more than 90%. This work represents the first time in the literature when the adverb *then* was investigated in such detail from a computational perspective, and the results are extremely promising.

More results from the evaluation of all the modules presented in this chapter are revealed in the following section.

4.4 Comparative evaluation for TE identification

This section presents detailed evaluation results for the TE identification task, obtained by decoupling the subtasks involved and illustrating the improvement in performance obtained after each processing stage. The evaluation is performed on the TERN 2004 training data released by the Linguistic Data Consortium (LDC) under catalogue number LDC2004E23 (Ferro et al., 2004). Attempts have been made to obtain the TERN 2004 test data to use it in the evaluation, but unfortunately this data is not publicly available as its release has not yet been approved. The TERN 2004 training data used in this work contains approximately 110,000 words annotated according to the TIMEX2 annotation guidelines presented in Section 3.2.2. The system performance on this corpus is measured using the official scoring script of the TERN competition (more details on the TERN data and competition can be found in Sections 3.3.1 and 3.4.5). The TERN scoring script compares the TIMEX2 tags from the system's output against the gold standard, evaluates each on a tag-by-tag basis, and produces summary metrics. Several settings are used for evaluation, settings that are described in detail below, followed by their evaluation results presented in Table 4.3.

4.4.1 Evaluation setting 1: rule-based identification only

The rule-based identification module is first evaluated on its own, to gain awareness of what performance a system can achieve by using only surface patterns that are context-independent.

4.4.2 Evaluation setting 2: setting 1 + syntactic correctness check

The output of the rule-based identification module is then checked for syntactic correctness, and a second evaluation is performed. Previous investigations have shown that only 90.2% TEs annotated in the TERN 2004 training data align exactly with a syntactic constituent, due to both parser and annotator errors (Ahn et al., 2007). This figure gives an estimated upper boundary on the recall of any method relying on syntactic constituency. A comparison between the results obtained before and after checking syntactic correctness shows a statistically significant increase in performance with a confidence level of 99%, both in the case of partial matches (TIMEX2), as well as when dealing with exact matches (TEXT).

4.4.3 Evaluation setting 3: setting 2 + annotation of anaphoric *then*

A third evaluation is concerned with the TE identification system incorporating both the module checking for syntactic correctness, as well as the module that disambiguates the occurrences of *then* and annotates only the anaphoric ones. It reveals a slight improvement in the results which is not statistically significant. This fact is explained by the relatively low number of anaphoric *then*s in comparison with the total number of TEs, representing only 0.4% of the TEs annotated in the gold standard. The improvement brought by the module dealing with the disambiguation of *then* to the overall results is approximately 0.1% both in the case of partial and exact matches. As part of this evaluation, all occurrences of *then* are disambiguated using the empirical approach described in Section 4.3.3,

and only the cases when it is used anaphorically receive a TIMEX2 annotation. Out of 80 occurrences of *then*, 73 are correctly classified as ANAPHORIC or NON_ANAPHORIC, so the classification is accurate in 91.25% of the cases. According to this classification, 21 occurrences of *then* are ANAPHORIC and receive a TIMEX2 annotation, and out of these 13 are correctly identified, while 8 are not annotated as TEs in the gold standard. Out of these 8 cases, 3 are errors made by the module that disambiguates *then* (as in [4.31]), while the other 5 occurrences of *then* should have received a TIMEX2 annotation, as they refer to specific points in time, but they were probably missed out by the annotators (as is the case in [4.32]). Two occurrences of *then* were wrongly classified as NON_ANAPHORIC, when they were annotated as TEs in the gold standard ([4.33]).

[4.31] *No marketing survey needed **then**.*

[4.32] *"The board elected to come up with a second-best answer in order to live to fight another day," said Dennis R. Beresford, an accounting professor at the University of Georgia who was **then** chairman of the accounting board.*

[4.33] *The bureau's statistics, **then**, were tabulated by a machine, the Remington Rand tabulator, a predecessor to the I.B.M.*

The detailed results obtained by the system in the three evaluation settings described above are presented in Table 4.3. The column **Possible** corresponds to the number of TEs annotated in the gold corpus, while the column **Actual** includes the number of TEs identified by the system. The columns **Correct**, **Incorrect**, **Missing**, **Spurious** indicate the number of TEs correctly identified, incorrectly matched, unidentified and over-generated by the system, respectively. **Precision** represents the number of correctly identified TEs divided by the

number of TEs present in the system output (**Precision = Correct / Actual**). **Recall** is the number of correctly identified TEs divided by the number of TEs annotated in the gold corpus (**Recall = Correct / Possible**). **F-measure** is calculated as the harmonic mean of precision and recall according to the formula: **F-measure = (2 · Precision · Recall) / (Precision + Recall)**.

The results show that the third evaluation setting that includes the module dealing with the disambiguation of *then* offers the best system performance. However, it should be noted that while the module checking for syntactic correctness significantly improves the results, the module dealing with *then* does not bring a statistically significant improvement to the overall results. The system output obtained in the best system setting is the focus of a detailed error analysis presented in the next section.

4.4.4 Error analysis

The errors generated by the system in its best setting (evaluation setting 3) are analysed using the output of the TERN official scorer, and they can be broken down in the following categories:

- **Incorrect extent:** there are 291 errors made in identifying the full TE extent (both the human annotators and the system identify the same TE, but the extent identified by the system does not fully match the one marked by the annotators);
- **Missing expressions:** there are 101 missing expressions, i.e. TEs annotated in the gold standard, but completely missing from the system output;
- **Spurious expressions:** there are 207 spurious TEs generated by the system but not annotated as TEs in the gold standard.

	Possible	Actual	Correct	Incorrect	Missing	Spurious	Precision	Recall	F-measure
Rule-based TE identification									
TIMEX2	3203	3431	3093	0	110	338	90.1%	96.6%	93.2%
TEXT	3203	3431	2617	476	110	338	76.3%	81.7%	78.9%
Rule-based TE identification checked for syntactic correctness									
TIMEX2	3203	3288	3089	0	114	199	93.9%	96.4%	95.2%
TEXT	3203	3288	2798	291	114	199	85.1%	87.4%	86.2%
Rule-based TE identification checked for syntactic correctness + anaphoric THEN									
TIMEX2	3203	3309	3102	0	101	207	93.7%	96.8%	95.3%
TEXT	3203	3309	2811	291	101	207	85.0%	87.8%	86.3%

Table 4.3: Evaluation results at different stages in the TE identification process

Incorrect extent

An analysis of the error cases marked as due to incorrect extent revealed that 79.72% are indeed system errors, while 20.78% are due to errors in the annotation of the gold standard. The largest source of errors is the wrong attachment of prepositional phrases (PPs). The expression *20 years in prison* is annotated as a TE in the gold standard, and the system only identifies a part of this expression *20 years*, due to the fact that the prepositional phrase *in prison* is not marked by the syntactic parser as being dependent on the noun phrase *20 years*. Another important error frequently made by the syntactic parser is the wrong attachment of determiners in longer NPs, as in the case of the determiner *a* from *a World War II-era mine* incorrectly linked to the noun *era* and therefore being included by the system in the TE *a World War II-era*, when only *World War II-era* represented the correct extent. Parser errors are also responsible for many incorrect TEs that contain appositive constructions or relative clauses (e.g. *an era in which Speakers have been defined by belligerent partisanship – particularly Mr. Gingrich and the Democrat he hounded from office, Jim Wright of Texas*). In these cases, the parser is either not able to link the appositive constructions to the main part of the expression that they modify, or the dependency structures it builds do not allow the correct identification of the relative clauses. This applies to the example above, for which the system fails to include the span of text *Jim Wright of Texas* in the recognised TE. Apart from the most frequent error sources enumerated above, one can encounter other cases of pre-modifiers or post-modifiers that are either omitted or wrongly included in a TE, such as the expression *this year* being identified when the correct expression would have been *this year alone*, or the expression *Odessa night* instead of the correct TE *night*. Besides all the errors

introduced by the syntactic parser, in 21.99% of the cases the system is not able to tag the correct extent of a TE due to modifiers not included in the lexicon and not syntactically dependent on the main trigger word(s) (e.g. from the annotated TE *the rest of the season*, the system only identifies *the season*), due to expression patterns not implemented (e.g. for *one hour every two weeks*, the system identifies two separate expressions *one hour* and *every two weeks*), and also due to errors in identifying certain range, conjoined and embedded expressions.

Missing expressions

The expressions labelled as missing from the system output account for 101 errors. The most frequent cause covering 26.73% missing TEs is represented by triggers not present in the lexicon. Certain words are intentionally not included in the lexicon due to their high ambiguity (e.g. *date*, *once*), others are infrequent words typically not associated with TEs (e.g. *heyday*, *workweek*). The second most frequent cause of missing TEs is the fact that certain expressions contain no temporal triggers, but they are either anaphoric or co-referential with another TE (e.g. ***It*** *is a date Armenians can point to with great pride.*). The lack of any temporal triggers is also to blame for missing numeric expressions that are ambiguous (e.g. *the 20th*). A rather interesting case is formed by expressions correctly identified by the rule-based TE identification module, but later discarded by the module checking for syntactic correctness. These are normally expressions that include a period (.) following an abbreviation, and the parser misclassifies it as marking the end of the sentence. The module checking for syntactic correctness does not allow TEs to extend across sentences, and the consequence is that valid TEs are discarded (e.g. *Aug. 17*). A number of errors appear due to unimplemented patterns, as is the case of the phrase *the next 24*

or 48 hours. The system is able to tag the expression *48 hours*, but misses the other expression *the next 24*.

Spurious expressions

A detailed analysis of the spurious cases revealed that 118 (57%) are due to legitimate TEs that are missing from the gold standard, and 89 are system errors. The largest proportion of spurious expressions is constituted by expressions from the class TOKEN, including occurrences of *now* (20), *former* (16), *future* (6), *recent(ly)* (6), etc. Other cases missed by the annotators include frequency denoting expressions (e.g. *annual*), expressions using the words *period* or *term*, as well as occurrences of *then*. There are also cases of expressions that are annotated in the gold standard, but appear as spurious due to the fact that they are part of a larger expression and the annotators did not respect the guidelines that clearly specify when two expressions appearing in close proximity should be independently tagged, and they combined two expressions into one. For example, the span of text *9 A.M. on Sept. 8, 1992* is wrongly annotated in the gold standard as one expression, while the present system follows the annotation guidelines and identifies two expressions, the consequence being that the scorer labels *9 A.M.* as spurious. The spurious TEs considered pure system errors include cases where temporal triggers are part of a Named Entity (e.g. *20th century* is part of the Named Entity *20th century fox*, and therefore is wrongly identified as a TE), expressions that are tagged individually even if they are part of larger expressions (e.g. the expressions *the first day* and *two-day* are tagged separately, despite the fact that they form one TE *the first day of the two-day summit*, therefore *two-day* is considered spurious), and expressions that include ambiguous trigger words (e.g. the word *quarter* is labelled as TE, but

the context *the quarter finals* dismisses a temporal meaning) or numbers that may denote years in different contexts (e.g. *LENGTH: 1964*). A large number of system errors (33.70%) involve the trigger word *time* (e.g. *this time*, *the next time*).

4.5 Adapting the system for TIMEX3-compliant TE identification

Given the aim of this thesis to cover the three main classes of temporal entities, and that the worldwide adopted standard for their annotation is TimeML (ISO-TimeML, 2007), the system developed for TIMEX2 annotation is now adjusted in order to perform TimeML-compliant TE annotation. As already mentioned in Section 3.2.4, TIMEX3 is the TimeML tag used for marking up temporal expressions.

This section describes the changes that the system undergoes at the temporal expression identification level to comply with the TimeML TIMEX3 annotation guidelines. The TimeML guidelines specify that the TIMEX3 tag should be applied to most TIMEX2 markable expressions. However, there are cases when the extent of the TIMEX3 markable expression differs from TIMEX2, with the main differences appearing in the case of embedded and post-modified TEs.

Embedded TEs are no longer allowed in TimeML, given a more general concept of temporal anchoring. The cases that required nesting according to the TIMEX2 guidelines are supposed to receive a different TIMEX3 annotation. The expressions that involve the use of temporal prepositions and conjunctions like *from*, *before*, *after* are in this situation. TIMEX3 requires that these connecting words are annotated as signals, and that temporal links should be used to capture

the relative ordering of the two TEs. Given the expression *two days before yesterday*, the TIMEX2 annotation would be the following:

```
<TIMEX2 VAL="2009-08-22">
    two days before
    <TIMEX2 VAL="2009-08-24">
        yesterday
    </TIMEX2>
</TIMEX2>
```

This expression receives a totally different TIMEX3 annotation captured below:

```
<TIMEX3 tid="t1" type="DURATION" value="P2D" beginPoint="t3"
endPoint="t2">
    two days
</TIMEX3>
<SIGNAL sid="s1">
    before
</SIGNAL>
<TIMEX3 tid="t2" type="DATE" value="2009-08-24"
temporalFunction="true" anchorTimeID="t0">
    yesterday
</TIMEX3> <TIMEX3 tid="t3" type="DATE" value="2009-08-22"
temporalFunction="true" anchorTimeID="t2"/>
```

As one can easily notice, these types of expressions are no longer considered to be calendar points as was the case in the TIMEX2 format, but anchored durations.

The example above features another important difference between the two annotation schemes. The temporal expression *t3* has no textual extent, but an empty TIMEX3 tag is inserted to substitute an expression relevant for interpreting a duration anchored by only one calendar point. Empty content TIMEX3 tags that do not consume any text represent a TE implicit in text.

Possessive constructions are also subject to change when moving from the TIMEX2 to the TIMEX3 annotation. They should no longer be annotated using two embedded tags, but they are supposed to be included in one TIMEX3 tag if both the possessive phrase and the phrase it modifies are temporal expressions. The expression *this year's summer* receives the following TIMEX2 annotation:

```
<TIMEX2 VAL="2009-SU">
    <TIMEX2 VAL="2009">
        this year
    </TIMEX2>
    's summer
</TIMEX2>
```

The corresponding TIMEX3 annotation is:

```
<TIMEX3 tid="t4" type="DATE" value="2009-SU">
    this year's summer
</TIMEX3>
```

The treatment given to post-modified TEs is another major difference between the two annotation standards. Post-modified TEs should no longer be annotated so that their extent includes the post-modifying phrase or clause. This applies to TEs that were previously annotated together with their post-modifiers.

These post-modifiers can be either relative clauses that describe a related event (examples [4.15] and [4.35] contrast the two annotation schemes), or prepositional phrases attached to the head of the TE (examples [4.36] and [4.37]). Both the relative clause *that Roosevelt died*, and the prepositional phrase *of experience* are to be excluded from the extent of the TIMEX3 tag, the markable expressions being now *the day* and *four decades*.

[4.34] *I remember <TIMEX2 VAL="1945-04-12">**the day that Roosevelt died**</TIMEX2>.*

[4.35] *I remember <TIMEX3 tid="t5" type="DATE" value="1945-04-12" temporalFunction="true" anchorTimeID="t6">**the day**</TIMEX3> that Roosevelt died.*

[4.36] *The company had <TIMEX2 VAL="P4DE">**four decades of experience**</TIMEX2>.*

[4.37] *The company had <TIMEX3 tid="t7" type="DURATION" value="P4DE">**four decades**</TIMEX3> of experience.*

These differences between TIMEX2 and TIMEX3 are mainly tackled by changes made in the module that deals with checking the syntactic correctness of TIMEX2 expressions. This was the module responsible both for the correct annotation of embedded expressions, and for the inclusion of post-modifiers in the extent of the TE. The changes cover all the differences described above. The system adaptation from TIMEX2 to TIMEX3 annotation was not difficult to implement, and this is the merit of the internal representation used by the system that was detailed and at the same time general enough to capture the semantics of each TE.

4.5.1 Results and error analysis

The adapted TIMEX3 TE identifier was evaluated on TimeBank 1.2 (Pustejovsky et al., 2006), the reference corpus annotated in compliance with the TimeML standard (please refer to Section 3.3.2 for more details). The results are obtained using the same scoring script employed by TERN 2004. In terms of partial matching, the system achieves an F-measure of 91.80%, while its accuracy in terms of exact matching is 86.70% (for more detailed results please refer to Table 5.6 included in Section 5.4). It is surprising to see that the inter-annotator agreement figure for the annotation of temporal expressions according to the TimeML standard is reported to be at around 83% (see Section 3.3.2), figure that is lower than the performance of the present system (86.70%). [2] A plausible explanation for this phenomenon would be the fact that the annotation that was performed on TimeBank is rather inconsistent, a fact that was also acknowledged by other authors (Boguraev and Ando, 2005, 2006), but also revealed during this system's error analysis process. A detailed system error analysis for the task of matching the exact extent of the TIMEX3 expression is presented below. This analysis is broken down into the same error categories as for TIMEX2 annotation: incorrect extent, missing expressions, spurious expressions.

Incorrect extent

The system's output for the TIMEX3 extent annotation task was compared against the TimeBank corpus using the scoring script which counted a number of 77 incorrect assignments. The analysis revealed that out of the 77 cases labelled

2. System performance was measured on the official release of the TimeBank 1.2 corpus through the Linguistic Data Consortium. Unfortunately the data used for measuring inter-annotator agreement are not readily available, and it is therefore impossible to provide an exact explanation.

by the scorer as incorrect, 47 are errors performed by the human annotators when labelling the corpus. The remaining 30 cases are errors made by the system. The errors performed by the human annotators are mainly due to the lack of clarity in specifying the annotation guidelines, especially in what the following issues are concerned:

- what and if premodifiers should be included in the extent of the TE. In many cases the determiner *the* is not present in the annotated TE (e.g. for *the fiscal second-quarter*, only *fiscal second-quarter* is annotated, for *the seasonally slow third quarter*, only *third quarter* is annotated);

- whether prepositions should be included in the extent of the TE. Annotators sometimes include the prepositions preceding a TE in the extent of the TE (e.g. *within 18 months* is annotated as a valid TE, when *within* should be annotated as a signal);

- how cases of expressions post-modified by *later*, *earlier* and *ago* should be annotated. Although such expressions manifest high degree of similarity both in the way they are built syntactically and in their semantics, they are annotated inconsistently: in many cases *later* and *earlier* are left outside the expression, while *ago* is included in the expression (e.g. the system annotates *a year earlier* as a TE, while annotators sometimes annotate this expression entirely, and in other cases they annotate *a year* as one TE, ignoring *earlier*, while throughout the corpus expressions including *ago* are consistently marked with *ago* included in the expression)

- sometimes annotators include punctuation marks in the extent of the expression (e.g. *the next few days.*)

- simple inconsistencies: sometimes the annotators include a premodifier in the extent of the TE (e.g. *later this afternoon*), sometimes not (e.g. in the case of *later this month*, the modifier *later* is not included in the extent of the expression).

If these cases were considered correct, then the system accuracy would go up to 89.81%.

As already mentioned, there are also 30 system errors, most of them (17) caused by errors of the syntactic parser that end up in attaching certain premodifiers that should not be present in the extent of the expression (e.g. due to the syntactic parser, the system identifies TEs such as *the company's new labor pact effective June 1*, *the invasion last August*). The rest of the errors (13) are due to expressions not covered by the implemented rules or lexicon (e.g. in the case of the TE *a good part of 1990*, only *1990* is annotated by the system).

Missing expressions

Apart from the cases labelled as incorrect by the scorer, there are also 31 cases of missing expressions that were present in the annotated corpus, but not present in the system's output. System errors account for 16 of the missing expressions, and they are mainly caused by the syntactic parser that splits sentences in the middle of an expression. Since the system's search for TEs is not performed across multiple sentences, parts belonging to TEs that are not marked by the system due to a sentence boundary present in the middle of the TE yield cases of missing TEs (e.g. a sentence boundary is inserted after *10 p.m* in *10 p.m. Wednesday*, thus preventing this part from being included in any TE). Missing TEs are also due to rules or words or expressions missing from the lexicon (e.g. *some time* is not captured by any rule, *moment* is not in the lexicon). The

other 15 cases of missing expressions are due to mistakes in human annotation, as they are expressions that were annotated when they should have not received an annotation. For example, there are expressions that despite the fact that they carry an intrinsic temporal value, no guidelines mention that they should be annotated, and sometimes human annotators mark them up (e.g. *meanwhile*, *already*, *yet*).

Spurious expressions

The scoring script has counted a very high number of spurious expressions (216), referring to expressions that are marked up by the system, but not labelled as such in the annotated corpus. An analysis of these cases revealed that 190 expressions should have been annotated by the human annotators, but were not. A high number of expressions (88) that were missed by the human annotators belong to the class TOKEN and include adverbs and adjectives such as: *now*, *former*, *future*, *current*, *currently*, *recent*, *recently*, *previously*, etc. Another class that accounts for many cases (39) not annotated in the gold corpus includes expressions denoting sets of times (e.g. *quarterly*). Another 26 expressions of type CALPOINT were missed by the annotators, mostly due to the fact that they were not sure whether generic usages of adverbs like *today* should have been annotated. Since the system follows the TIMEX2 guidelines in the cases where the TimeML guidelines are under-specified, these cases are identified by the system. It is clear that the annotators were confused about how to annotate such generic cases, as sometimes they are annotated and in other cases they are not. If these 190 spurious expressions would have been annotated in the gold corpus, the system accuracy would have been 93.40%.

Besides these cases missed by human annotators there are also 21 TEs present

in article headers and footers that were not annotated, and the system correctly identifies them (e.g. *02-13-98 1426EST*).

The remaining 26 cases are pure system errors. Many of them (10) are cases of durations (mainly ages) that the system marks up because the author considers that the annotation of ages could prove useful to a system that reasons about time (e.g. *52 years old*). There are also 7 errors made by the system when annotating TEs mentioned as part of proper names (e.g. *This Week* is annotated by the system in the context of *ABC-TV's "This Week With David Brinkley"*). The rest of the system errors are due to metaphorical usages that the system cannot identify (e.g. *the eve* in the context *the eve of the return to peace talks*), or cases of coordinations or disjunctions of TEs annotated separately (e.g. *recent weeks and months* is annotated as one TE in the corpus and the system generates two expressions *recent weeks* and *months*), or cases that the system identifies due to the presence of lexical triggers, but which probably are not supposed to be annotated (e.g. *a reasonably flat year*, *a matter of days*, *the transition period*).

This section has shown that a TIMEX2 annotation system can be adapted to perform TIMEX3 annotation, and the results obtained for both annotation types are comparable. The error analysis revealed that the number of system errors is relatively low compared to the cases correctly identified by the system, but incorrectly marked by human annotators. This can be seen as an indication that the annotation guidelines could be consistently revised and improved.

4.6 Conclusions

The aim of this chapter was to illustrate how the problem of TE identification has been addressed in the context of this research.

The first part of the chapter introduced a detailed classification of the temporal expressions targeted by existing annotation schemes and systems performing temporal expression annotation.

The development process of a system aiming at automatically identifying all types of temporal expressions was then described. The automatic TE identification system presented in Section 4.3 relies on several modules, each one providing extra knowledge: the rule-based TE identification module, the module that checks for syntactic correctness, and the module that disambiguates the occurrences of *then*. After describing these modules in detail, a comprehensive evaluation of the results obtained after adding each knowledge source to the TE identification algorithm is captured by Section 4.4.

The changes required to adapt the system from the TIMEX2 annotation standard to the TIMEX3 specification are described in detail in Section 4.5. This section also includes a detailed evaluation and error analysis of the TIMEX3 TE identifier. The TIMEX2 to TIMEX3 adaptation process stands as proof that the representation used by the system is general enough to be adapted to any TIMEX standard, should any other TE annotation standard be introduced in the future.

As already mentioned on several occasions, the temporal expression annotation process is completed only when each temporal expression is assigned a series of attributes and attribute values in accordance with a chosen annotation scheme. This is done at the normalisation stage whose description can be found in the next chapter.

Chapter 5

Temporal Expression Normalisation

5.1 Overview

The process of temporal expression annotation comprises two stages: the TE identification stage and the TE normalisation stage. Chapter 4 described in detail the first stage whose output was the set of temporal expressions identified in text, along with feature-typed structures that embed information about a TE's internal semantic content. This information extracted at the identification stage is exploited at the normalisation stage in order to find the value that a certain expression designates or is intended to designate, value that is sometimes dependent both on the TE's internal semantic content and on context-dependent factors.

This chapter focuses on the normalisation stage of the TE annotation process. **Normalisation** (or **temporal resolution**) is the whole process carried out to identify the final values of the attributes attached to a temporal expression. These attributes depend on the annotation scheme used. During normalisation the values of the attributes can either be extracted from the expression itself, or

calculated using the attribute values of another TE which serves as anchor time.

The result obtained by normalising a temporal expression is spread across various attributes that characterise the TE according to the chosen annotation scheme. For evaluation purposes, the TIMEX2 annotation scheme is initially adopted, due to its high level of detail and complexity among existing schemes. Different normalisation models are experimented with, and a detailed description of each model can be found in Section 5.2. A comparative evaluation of these alternative models for TE normalisation is captured in Section 5.3.

The best performing TE normalisation module developed for TIMEX2 annotation is then adapted to the TIMEX3 annotation scheme, part of the TimeML standard, and another evaluation is performed on TimeBank. The changes involved in this process, as well as the results obtained by evaluating the TIMEX3-adapted normaliser are described in Section 5.4. This section also includes a detailed analysis of the errors and problems encountered during the TIMEX3 annotation process. The chapter finishes with conclusions.

5.2 Methodology for the normalisation of temporal expressions

Temporal expression normalisation is the process carried out in order to identify the values of the attributes attached to every TE. According to the TIMEX2 guidelines, one or more attributes should be assigned to a TE, and these attributes are: VAL, MOD, ANCHOR_VAL, ANCHOR_DIR, and SET. Their usage was described in detail in Section 3.2.2, and can be found in a summarised form in Table 5.1 (adapted from the TIMEX2 guidelines).

In this research the values of the MOD, ANCHOR_DIR and SET attributes

Attribute	Function	Example
VAL	Contains a normalised form of the date/time, duration or set of times denoted by the expression.	VAL="2000-10-15"
MOD	Captures temporal modifiers.	MOD="APPROX"
ANCHOR_VAL	Contains a normalised form of the date/time a TE of type DURATION or TOKEN is anchored to.	ANCHOR_VAL="2000-10-15"
ANCHOR_DIR	Captures the relative direction or orientation of the period of time denoted by the TE of type DURATION or TOKEN with respect to the ANCHOR_VAL.	ANCHOR_DIR="BEFORE"
SET	Singles out expressions referring to sets of times.	SET="YES"

Table 5.1: TIMEX2 attributes and their usage

are determined at the rule-based TE identification stage presented in the previous chapter, and then included in the semantic representations generated for a TE at the time of matching. Therefore, the TE normalisation stage presented here only focuses on establishing the values of VAL and ANCHOR_VAL. To do this, the same process is used to determine the temporal anchor, which then is used fill in either the value of VAL or ANCHOR_VAL, depending on the nature of the expression. The set of TEs that require a normalised value to be assigned to their VAL attribute includes underspecified CALPOINT TEs and is disjunct from the set of TEs that require a value for their ANCHOR_VAL attribute and that includes expressions of type DURATION and TOKEN. The difference between the two sets is made by the way the temporal anchor is used. For an element of the first set, the anchor's value of the VAL attribute serves as argument for the temporal function assigned to that element. In the case of the second set, the anchor's value of VAL is coerced to the required granularity and the resulted value is assigned to ANCHOR_VAL.

For many TEs, the semantic representations built at the identification stage can be directly translated into a normalised value for the attribute VAL. This is the case of fully specified calendar points, durations and TEs of type FREQUENCY, TOKEN or UNANCHORABLE. However, the remaining CALPOINT TEs that are under-specified require a temporal anchor for computing the final normalised value of their VAL attribute. In the case of the attribute ANCHOR_VAL, only those TEs of type DURATION and TOKEN that were assigned a value for the attribute ANCHOR_DIR should also receive a value for ANCHOR_VAL.

The temporal anchor is a temporal expression typically mentioned earlier in text whose value is specified up to the level of granularity required to interpret the underspecified expression. In the case of EMBEDDED TEs, the temporal anchor for the embedding TE is always the embedded expression. For all remaining TEs, the anchor can be determined using several tracking models. A number of temporal anchor tracking models have been experimented with, all having different levels of context dependency. Their description can be found below.

5.2.1 Norm-DCT: Normalisation with respect to the Document Creation Time

The most frequent heuristic employed by researchers in normalising TEs is to choose the document timestamp as the temporal anchor for all under-specified temporal expressions. Since TE annotation is typically applied to news articles that have a precise date assigned to them, a straightforward way of doing normalisation is to consider that all underspecified temporal expressions are relative to the time of the article.

Every temporal expression whose value for the attribute VAL should be filled using a temporal function is normalised by using the Document Creation Time (DCT) as argument for the function. Temporal calculations are performed by making use of a freely available package [1] for date arithmetics based on the Gregorian calendar.

There are certain issues that appear during the normalisation process. One issue is related to which named part of a cycle (a cycle that can either be a week or a year) does one refer to when using expressions like *last/next* followed by a named TE providing a position in a cycle (e.g. *last Tuesday*, *next summer*). Considering the TE *last Tuesday*, it is not clear which larger-granularity cycle should be chosen (*this week* or *last week*) if the anchor's position in the cycle (in this case that particular week) is later than *Tuesday*. If an expression like *last Tuesday* is used on a *Friday*, one could have referred to the Tuesday belonging to the same week (in this case *last* is used only to highlight that the day is located in the past with respect to the DCT), or the reference could have been to the *Tuesday* of *last week*. This normalisation model assumes the latter usage and dismisses the possibility that such expressions refer to a time point situated in the same cycle as the temporal anchor. The model described in Section 5.2.5 takes this possibility into account and performs a more complex processing of these cases.

Another issue is related to the context dependency of the semantic class of the TE. Certain expressions can manifest semantic class ambiguity, in the sense that they can refer either to a time point, to a duration or to a frequency, and the usage is selected by the context. In the examples below, the expression *a*

1. The Date::Calc package is used for all Gregorian calendar date calculations. It is a Perl module freely downloadable from http://www.cpan.org/.

year is in the first case a frequency ([5.1]), in the second case a simple duration ([5.2]), and in the third case a dependent calendar point (([5.3])) situated a year later than the temporal anchor.

[5.1] *He makes $20K profit* ***a year****.*

[5.2] *He has lived in London for* ***a year****.*

[5.3] *He will finish his degree in* ***a year****.*

The disambiguation is performed by using the contextual information given by the prepositions that precede the TE. Prepositions like *in* and *within* indicate a dependent calendar point, while prepositions like *for* or *during* trigger a duration. The unclear cases are considered either frequencies (this applies to expressions of the type *a + UNIT*, where UNIT is either a time or date unit) or durations (this applies to expressions encoding a number of temporal units greater than one, such as *three months*).

For those TEs that already have a value assigned to the VAL attribute, but due to the fact that their semantics led to assigning a value to the ANCHOR_DIR attribute, the ANCHOR_VAL attribute needs to receive a value as well. This is filled by coercing the DCT to the granularity of the expression, if this granularity is explicitly mentioned. Given the example [5.4], the expression *the past three years* is characterised by the value *ENDING* for the attribute ANCHOR_DIR, and ANCHOR_VAL receives the value of the DCT (supposedly *19/04/1996*) coerced to the granularity of the expression (i.e. year), thus *1996*.

[5.4] *She has lived in Spain for* ***the past three years****.*

Despite the fact that this normalisation model proves efficient in the case of news stories because they are relatively short and the events are temporally located in the immediate vicinity of the DCT, one must acknowledge that the temporal focus changes as the discourse progresses, and that not all TEs

should be treated equally in terms of choosing their temporal anchor due to the dependent vs. deictic distinction. Therefore several other normalisation models are experimented with below to find ways to improve the normalisation process.

5.2.2 Norm-Recent: Normalisation with respect to the most recent suitable TE

The second normalisation model is adopted from the field of Anaphora Resolution (Mitkov, 2003) where the distance between the anaphoric pronoun and a possible candidate is a good indicator of how likely it is that the candidate is the antecedent of the pronoun. This hypothesis leads to the idea of using the most recent TE mentioned in text as temporal anchor for under-specified TEs. This recency-based model relies on a linear list of all the temporal expressions mentioned so far in the text, a list that is ordered by recency. For each under-specified expression, the temporal anchor is chosen to be the most recent TE in the list that refers to a calendar point and is fine-grained enough to comply with the granularity required by the under-specified TE. This is equivalent to considering that all under-specified TEs should be interpreted with respect to Reichenbach's Reference Time Point (see Section 2.5.4), and that all fully specified or already resolved TEs modify this Reference Time.

This recency based model does not account for the distinction between deictic and dependent under-specified expressions, thus failing in providing the correct interpretation for deictic TEs. The model described in Section 5.2.3 solves the problem of deictic expressions by anchoring them to the Speech Time, which in the case of news articles is the same as the Document Creation Time.

5.2.3 Norm-Class: Backward looking class-sensitive normalisation

The distinction between deictic, dependent and flexible anchoring TEs is relevant for the normalisation process for reasons already mentioned in Section 4.2. The time expressed by a deictic TE is relative to the Speech Time Point, and in the case of news articles the Speech Time Point is readily available as the Document Creation Time (DCT). Dependent TEs should be anchored to the Reference Time Point that is most prominent in the preceding discourse. Flexible anchoring TEs can either be anchored to the Speech Time Point or to the Reference Time Point. Corpus investigation done as part of this research has revealed that in most cases they are anchored to the Speech Time Point, and therefore from this point forward they will receive the same treatment as the deictic TEs.

Since it is relatively easy to automatically distinguish between deictic and dependent TEs, the present normalisation model takes advantage of this information and combines the two heuristics previously used independently as part of the Norm-DCT and Norm-Recent normalisation models. Deictic TEs are now normalised by using the DCT as temporal anchor, while TEs classified as DEPENDENT are normalised with respect to the most recent TE mentioned in text whose value is fully specified down to the granularity of the expression to be resolved. This normalisation method chooses the temporal anchor for dependent TEs from the set of already resolved or fully specified TEs in the reverse order to the way they are mentioned in text.

This heuristic that chooses as temporal anchor for a dependent TE the most recent previous TE of suitable granularity represents a rather simplified view of how the Reference Time Point, also referred to as temporal focus (Webber,

1988), is instantiated throughout the discourse. The heuristic is equivalent to considering that every calendar point TE modifies the Reference Time and is accessible for future reference. While counterexamples to this rule can easily be found in real text and discourse (example [5.5]), it represents only an approximate solution until a better understanding of how the temporal focus evolves during discourse and how this can be modelled automatically becomes available.

[5.5] *John finished on* ***Wednesday*** *the volume he started reading on* ***Monday***. ***Three days later*** *he finished another volume.*

The problem of finding an appropriate anchor for a given TE is very much influenced by where one looks for this anchor. This can be likened to the problem of knowing at each point in the discourse which is the domain of referential accessibility, i.e. in what part of the discourse should the anchor be situated. For this problem, different strategies could be imagined.

A straightforward choice is considering that the entire previous discourse is the domain of referential accessibility, without establishing which TEs are closed to being referred to, and considering as candidate anchors all the TEs found in text in the linear order they appear. The chosen anchor would be the most recent TE having a suitable granularity. Both the Norm-Recent and Norm-Class models have adopted this approach.

Other strategies for defining the domain of referential accessibility could be inspired from theories of discourse structure that define accessibility in the current discourse unit either using **attentional states** (Grosz and Sidner, 1986), or **veins theory** (Cristea et al., 1998). **Attentional states** are abstractions of the focus of attention of the participants as the discourse unfolds. They summarise information about objects, properties and relations that are most

salient in previous utterances, information that is considered crucial for processing subsequent discourse. **Veins theory** is a generalisation of **centering theory** (Grosz et al., 1995) that delimits domains of referential accessibility for each unit in a discourse by exploiting rhetorical relations between nuclei, considered essential for the writer's purpose, and satellites that increase understanding, but are not essential in discourse. One could employ either one of these discourse structure theories for constraining the set of discourse units where the anchor for a given underspecified TE should be located, but such an approach would be suitable only for theoretical studies, as they do not represent a feasible choice from the perspective of implementing automatic systems due to their extensive use of semantic information.

Due to the difficulty and complexity of achieving a correct semantic approach, the following normalisation model tries to define accessibility in the current discourse unit (considered to be the current clause) using information that is readily available as a result of the automatic processing performed so far by the system.

5.2.4 Norm-Local: Class-sensitive normalisation prioritising clause-local context

In an attempt to define the accessibility domain of each TE situated in a given syntactic clause, the previous temporal normalisation model is enhanced with the following heuristic: only the fully specified, deictic and flexible TEs present in previous clauses are included in the accessibility domain of a given TE. At the same time, the accessibility domain of a given TE is enriched by adding all flexible and dependent TEs of coarser granularity situated in the same clause, irrespective

of their relative position in the clause. In this way, dependent TEs are considered less prominent in discourse and they can only be included in the accessibility domain of a TE located in the same clause, and also the possibility arises of anchoring a TE to an expression that is not present in the previous discourse, but mentioned after the TE to be normalised. This accounts for the theoretical observation that in a clause where an adverbial modifies the Event Time and another one the Reference Time, the Reference Time-modifying adverbial must occur after the Event Time-modifying adverbial, since the Reference Time-modifying adverbial is structurally above the Event Time-modifying adverbial (Thompson, 2005).

In this normalisation model, all TEs present in a clause are normalised starting from the most coarse-grained one to the one with the lowest granularity irrespective of their order in text. In the case of a dependent or flexible TE, the anchor is located in its accessibility domain defined as above, by searching first the expressions from the same clause and then the fully specified, deictic and flexible expressions found in the preceding discourse. For deictic TEs the anchor is considered to be the DCT, as in the previous model.

Sentence [5.6] is used to illustrate this model. In this example, the first expression to be normalised is *Monday* and its anchor is the DCT. The next expression to be normalised is *9:30 a.m.* and its anchor would be correctly determined by this model as being *Monday*, an already resolved TE located in the same clause.

[5.6] *The meeting is at* ***9:30 a.m.*** *on* ***Monday****.*

Clearly, priority is given to the TEs situated in the same clause as the TE to be normalised, and if no suitable anchor TE for dependent or flexible expressions is found in the same clause, the search is conducted in preceding discourse. For

deictic expressions the anchor is considered to be the DCT .

Sections 5.2.1 to 5.2.4 have presented four different normalisation models that attempt in different ways to find the most appropriate temporal anchor for a given TE. But finding the anchor is not the only context-dependent problem one must solve to be able to correctly normalise an under-specified TE. Another context-dependent ambiguity that arises for certain expressions is concerned with the direction of the relation between a referential TE and its anchor. This is called **the direction problem** and is described in more detail in the following section.

5.2.5 The direction problem

A problem that often appears during normalisation is not knowing what cycle is meant when using a named expression like *Thursday* or *October 15* without any direction indicator such as *last* or *next*. This is known as the **direction problem**, and solving it involves disambiguating the direction intended in the utterance, and more specifically which cycle should be chosen among the one immediately before the cycle containing the temporal anchor, the cycle containing the temporal anchor, or the cycle immediately following it. In the case of a TE at day-level granularity such as *Thursday*, the problem consists in finding the exact week to which this specific day belongs to. One should decide using the context whether the author refers to the Thursday belonging to the same week as the temporal anchor, or to the Thursday belonging to the previous or the following week.

[5.7] *I have a doctor appointment on* ***Thursday***.[2]

[5.8] *I had a doctor appointment on* ***Thursday***.[3]

2. If uttered on a Monday, it is the Thursday of the same week, but if uttered on a Saturday, then the Thursday belonging to the following week is intended.

3. If uttered on a Saturday, the same-week interpretation should be given, while if the

In the examples above, the factors that appear to be relevant for choosing the right interpretation are the tense of the verb the TE depends on, and the relative position in time of the TE with respect to the temporal anchor. However, there are cases that these factors can not predict the correct behaviour for, and deeper semantic understanding is needed.

[5.9] *I wanted to go swimming on* ***Thursday****.*

The ambiguity of example [5.9] is highlighted in [5.10] and [5.11]. In the context of example [5.10], the closest Thursday preceding the temporal anchor should be chosen, but in contexts like [5.11] it is very difficult to automatically predict that the speaker refers to the closest Thursday following the temporal anchor. Such cases are currently tackled in this work by using the same factors presented above (i.e. the verb tense and the relative position with respect to the anchor), and no attempt to understand the verb semantics is made, thus both TEs from examples [5.10] and [5.11] receive the same interpretation that only proves correct in the case of the former sentence.

[5.10] *I wanted to go swimming on* ***Thursday****, but I had too much work.*

[5.11] *I wanted to go swimming on* ***Thursday****, but I will have to cancel.*

Ahn et al. (2007) attempt to model the direction problem as a classification problem with features including verb tense and lexical features for a context window of 3 words. The targeted classification classes are *SAME*, *FORWARD* and *BACKWARD*. Confronted with the direction problem, Mani and Wilson (2000) use rules that look at the tense of the closest verb in the same clause as the TE to predict a direction. They only deal with day names and do not target other named TEs, such as season names, references to fiscal quarters and date-month expressions for which the unknown cycle is the year to which they belong.

---

temporal anchor is a Monday, the Thursday of the previous week should be chosen.

It is true that in most news articles the period of time talked about is typically in the immediate neighbourhood of the DCT, and the intended year is the same as the DCT year, but there are also cases when the same factors discussed above can prove useful in deciding the correct year for an expression.

The present work accounts for all named expressions for which the cycle is ambiguous. The main factors contributing towards choosing a cycle for a named TE are the tense of the verb the TE syntactically depends on, and the relative temporal position of the TE with respect to the temporal anchor. If no tense information is present in the sentence that includes the named TE, three calendar point variants are generated to correspond to the three choices of temporal cycle, and having the details of the named TE. The distance between each variant and the temporal anchor is measured as being the number of days separating the two dates. The variant closest to the temporal anchor is chosen.

When the tense information of the verb modified by the named TE can be identified, only the distinction past versus non-past is relevant. Unlike other systems that use the tense of the first tensed verb located in the same sentence as the named TE (Ahn et al., 2005b), the present work relies on the verb the TE is dependent on due to the availability of the syntactic dependency relations provided by Connexor's FDG parser.

At this point, given either a past or a non-past tense for the verb modified by the TE, the relative temporal position of the named TE with respect to the temporal anchor becomes relevant. For a past tense verb, the most recent time point prior to the temporal anchor should be chosen. If the anchor's position in the cycle is later than the position denoted by the underspecified TE, the same cycle is chosen, otherwise the previous cycle is considered more appropriate. For a non-past tense verb, the closest time point situated in the future with respect to

the temporal anchor and corresponding to the named TE's description constitutes the solution. The same cycle is chosen when the anchor's position in the cycle is earlier than the position denoted by the TE, and the following cycle applies to the remaining cases.

After having decided on a cycle, the final value is produced for the VAL attribute of the investigated TE.

Another context dependent problem that needs to be solved at the normalisation stage is concerned with setting apart generic from specific usages of certain temporal adverbs like *today*. This is known as **the generic vs. specific problem**, and is discussed in detail in the following section.

5.2.6 The generic vs. specific problem

A well-known issue in the area of TE normalisation is being able to distinguish between specific and non-specific readings of adverbials like *today*, *yesterday* and *tomorrow*. Being able to make this distinction would prove useful for annotating specific usages of these adverbs with an exact date (in this case they would belong to the class of DEICTIC CALPOINT TEs), and generic usages with token values (in this case they would belong to the class of TOKEN TEs) indicating generic reference to the past (e.g. ***yesterday**'s music* => PAST_REF), present (e.g. ***today**'s youth* => PRESENT_REF) or future (e.g. ***tomorrow**'s engineers* => FUTURE_REF). For generic references to the present moment, the attribute ANCHOR_DIR should receive the value AS_OF, for past references it should be set on BEFORE, and for future references the value should be AFTER. If the attribute ANCHOR_DIR is assigned a value, the attribute ANCHOR_VAL should also receive a value, and for TOKEN TEs this value is filled using the DCT.

Mani and Wilson (2000) present a machine learning approach for determining whether an occurrence of the word *today* refers specifically to the day of the utterance or generically to the present. The authors then hand-coded the most prominent rules learnt by the classifier in a module that performs the disambiguation. Only a few features described by Mani and Wilson (2000) are used in the present research, and interestingly enough none of the features their classifier found extremely relevant proved useful here (e.g. presence of the word *most* in the same sentence). This could be due to the fact that rules induced automatically using machine learning techniques are sometimes opaque, too specific (e.g. the word *most* is present only in a few cases) and perhaps overfitted. In order to overcome these shortcomings, this approach relies on heuristics inspired by grammatical rules, as generic rules justified by the English grammar are thought to perform better, especially in the case of a generic automatic system.

The present approach taken for solving the generic vs. specific problem relies on two simple rules. The first rule predicts generic usage if the tense of the governing verb phrase is Present Tense Simple (usually employed in generic contexts) and the subject corresponding to this VP is generic (generic subjects are considered to be bare plurals, the pronoun *it* and the adverb *there*). The second rule is also for detecting generic usage, but this time targets possessive constructions (e.g. ***yesterday****'s music*, *the youth of* ***today***). All the cases not satisfying any of these rules are considered to be specific mentions. Following this disambiguation process, each case is annotated accordingly: specific usages are normalised according to the adverb's corresponding function taking as argument the DCT, while for generic usages the attributes ANCHOR_VAL and ANCHOR_DIR are filled as described above.

After describing the four proposed methods of identifying the temporal anchor for under-specified TEs in Sections 5.2.1 to 5.2.4, and the solutions implemented for the direction and the generic vs. specific problems in Sections 5.2.5 and 5.2.6 respectively, the focus is now on answering the following research questions: what is the best model for finding the temporal anchor, and what is the impact of solving the direction and the generic vs. specific problems on the entire normalisation process? The next section describes the experiments performed in trying to answer these research questions, and the results achieved.

5.3 Comparative evaluation of TE normalisation methods

This section captures detailed evaluation results for the task of TE normalisation. The four temporal anchor tracking models presented in sections 5.2.1 to 5.2.4 have been evaluated in turn, to reveal the best approach for identifying the anchor for an under-specified TE. Following this evaluation, the best performing model is chosen and modules are added to deal with the two major problems that appear during normalisation: the direction problem and the generic vs. specific problem. Two more evaluations are performed after adding each module to reveal the contribution brought by each one of them.

As in the case of TE identification, the evaluation is performed on the TERN 2004 training data, using the official scoring script of the TERN competition. For each targeted attribute, this script looks only at those TEs from the system output that partially match TEs annotated in the gold corpus (i.e. those counted in the cells corresponding to CORRECT TIMEX2s in Table 4.3). This means that those TEs missed or over-generated by the system (columns Missing or Spurious

from Table 4.3) are ignored for the purpose of evaluating system performance on attribute values. It should be noted that the extent of a system identified TE does not need to match exactly the corresponding human annotated TE in order for its attribute values to be evaluated against the gold standard. For each attribute included in the TIMEX2 tag, the scoring script calculates the same figures encountered at the TE identification stage (Section 4.4), figures that correspond to **Possible**, **Actual**, **Correct**, **Incorrect**, **Missing**, **Spurious** attribute values, and figures that indicate the **Precision**, **Recall** and **F-measure** of the system in assigning attribute values. Given the attribute VAL for example and looking only at those TEs identified by the system that partially or fully match a corresponding TE in the gold corpus, the TERN script counts the following:

- the number of VAL attribute values found in the corpus for these TEs (**Possible**);

- the number of VAL attribute values assigned by the system to these TEs (**Actual**);

- the number of correct and incorrect VAL assignments made by the system (**Correct** and **Incorrect**, respectively). When comparing attribute values and making the correct vs. incorrect decision, only exact matching between the value assigned by the system and the value annotated in the corpus leads to considering an assignment correct. No fuzzy matching is used when comparing attribute values.

- the number of VAL attribute values missing from the system output and present in the gold corpus (**Missing**);

- the number of VAL attribute values assigned by the system to TEs that have no associated VAL in the gold standard (**Spurious**).

The figures that indicate the precision, recall and F-measure of the system in assigning attribute values are computed using the same formulae as at the TE identification stage:

Precision = Correct / Actual

Recall = Correct / Possible

F-measure = (2 · Precision · Recall) / (Precision + Recall)

As previously mentioned, the normalisation stage focuses only on filling the values of the VAL and ANCHOR_VAL attributes. This comparative evaluation will therefore present the changes in system performance for these two attributes. However, in the following, the analysis will mainly focus on the VAL attribute, not only because it is the most important of all TIMEX2 attributes, but also due to the high number of annotation errors and inconsistencies encountered among the values of the ANCHOR_VAL attribute.

The complete results of the normalisation models described in the previous section are presented in Table 5.2. The table also includes results corresponding to a baseline presented in detail below.

To monitor the benefits brought by each normalisation model, a baseline model was considered: **Norm-Baseline**. In the case of the VAL attribute, the baseline model only assigned a VAL attribute to the fully specified expressions that embedded their full value and did not require recourse to any context-dependent processing. As part of the baseline model, the ANCHOR_VAL attribute always received the value of the DCT whenever the attribute ANCHOR_DIR was assigned a value. This baseline model achieved an F-measure of 60.8% for VAL, and 35.6% for ANCHOR_VAL.

	Possible	Actual	Correct	Incorrect	Missing	Spur	Precision	Recall	F-measure
Norm-Baseline: Baseline normalisation									
VAL	2983	2459	1654	793	536	12	67.3%	55.4%	60.8%
ANCHOR_VAL	637	530	208	266	163	56	39.2%	32.7%	35.6%
Norm-DCT: Normalisation with respect to the DCT									
VAL	2983	2929	2519	390	74	20	86.0%	84.4%	85.2%
ANCHOR_VAL	637	530	346	128	163	56	65.3%	54.3%	59.3%
Norm-Recent: Normalisation with respect to the most recent suitable TE									
VAL	2983	2928	2442	466	75	20	83.4%	81.9%	82.6%
ANCHOR_VAL	637	530	214	260	163	56	40.4%	33.6%	36.7%
Norm-Class: Backward looking class-sensitive normalisation									
VAL	2983	2929	2539	370	74	20	86.7%	85.1%	85.9%
ANCHOR_VAL	637	530	388	86	163	56	73.2%	60.9%	66.5%
Norm-Local: Class-sensitive normalisation prioritising clause-local context									
VAL	2983	2929	2547	362	74	20	87.0%	85.4%	86.2%
ANCHOR_VAL	637	530	388	86	163	56	73.2%	60.9%	66.5%
Norm-Local with direction disambiguation									
VAL	2983	2929	2596	313	74	20	88.6%	87.0%	87.8%
ANCHOR_VAL	637	530	391	83	163	56	73.8%	61.4%	67.0%
Norm-Local with direction disambiguation and generic vs. specific distinction									
VAL	2983	2929	2602	307	74	20	88.8%	87.2%	88.0%
ANCHOR_VAL	637	544	399	85	153	60	73.3%	62.6%	67.6%

Table 5.2: Comparative evaluation results for different normalisation models

The first temporal anchor tracking model evaluated is the one that considers as temporal anchor for all TEs the DCT, i.e. the **Norm-DCT** model. The results are promising, despite the simplicity of this anchor tracking model.

Norm-Recent is the second temporal anchor tracking model evaluated. It consists in using the most recent TE mentioned in text as temporal anchor for all under-specified TEs. The results of this model are statistically worse than the ones of the first model [4]. In the case of the VAL attribute, there are 500 expressions denoting calendar points that are classified as deictic, 47 classified as dependent, and 642 as flexible in terms of anchoring. It is interesting that, by considering all TEs as being dependent, and most expressions in the corpus being either deictic or flexible, the number of incorrect values assigned to VAL raises with only 76 compared to when considering all expressions to be deictic. This confirms the observation that in newswire articles the reference time rarely shifts from the document creation time, as most events described in an article are located temporally in the immediate vicinity of the DCT.

The model evaluated next is **Norm-Class**. It combines the Norm-DCT and Norm-Recent models and accounts for the distinction between deictic and dependent TEs. The Norm-Class model considers that the time expressed by a deictic TE is relative to the DCT, and that dependent TEs are relative to the most prominent Reference Time point introduced in the preceding discourse. In this model the most prominent Reference Time point is considered to be the most recent TE mentioned in the text having a suitable granularity for the expression to be resolved. When comparing this model with the one that normalises all TEs with respect to the DCT, an improvement in the value of VAL can be seen for 20 expressions. Considering that only 47 expressions are dependent, the

4. The test of significance used is t-test, and the confidence level is 99%.

maximum number of changes expected when changing to a normalisation model that only affects dependent expressions is 47. A detailed analysis of dependent TEs reveals that 21 cases are occurrences of *then*, the other 26 being distributed across the other subclasses of dependent TEs denoting calendar points. There are 8 occurrences of *then* that are not annotated in the gold standard, therefore it is impossible to obtain an improvement for their VAL attribute, as they will always be counted as spurious cases. From the 13 (21-8) remaining *then*s, 3 are pointing to an event, making it almost impossible for a system to assign them a correct value. This leaves room for improvement in 10 occurrences of *then*. All these 10 occurrences were assigned an incorrect value for VAL by the Norm-DCT model. It is surprising to find that in 9 cases out of 10, the current Norm-Class model assigns the correct value for the corresponding VAL attribute. When looking at the other 26 dependent TEs, 7 of them are cases that an automatic system would not be able to assign a correct value to without access to world knowledge and capability for advanced reasoning, as they are either event-anchored, or non-specific, or refer to a period of time in the past that is not clearly delimited. From the remaining 19, 4 are assigned a correct value for VAL by the Norm-DCT model, while 15 are assigned incorrect values. The 4 correct ones are also assigned a correct value using the Norm-Class model, and in addition this model also assigns the correct value to another 11 TEs out of the 15 previously incorrect cases. One can conclude that this model is appropriate for dependent TEs, as in 25 (10+15) automatically correctable cases, it manages to assign the correct value for 20 (9+11) TEs.

The fourth temporal anchor tracking model evaluated is the **Norm-Local** model. It differs from the previous model through the fact that it defines in a novel manner the accessibility domain of each TE and prioritises clause-local

context. Unlike any previous models, the Norm-Local model allows cataphoric-like bridging relations to be established between the expression to be resolved and its anchor. It considers that all TEs can serve as anchors for other TEs of finer granularity situated in the same clause. This model also limits the influence of dependent TEs on subsequent discourse, in the sense that a dependent TE cannot serve as anchor for a TE situated in a different clause. In addition, the Norm-Local model allows flexible TEs to be resolved using a coarser-grained TE situated in the same clause, unlike the previous model that was anchoring all flexible TEs to the DCT. When compared with the Norm-Class model, the current model helps in solving 12 more TEs correctly, but at the same time introduces 4 errors.

According to the results obtained so far, the best performing temporal anchor tracking model is considered to be the Norm-Local model, the class-sensitive normalisation model prioritising clause-local context. This model is further enhanced with a module that deals with **the direction problem** as described in Section 5.2.5. The results are significantly better, with 69 previously incorrect valued TEs now receiving a correct value. However, 20 errors are introduced at this stage, some due to parser errors that led to assigning a wrong tense description to the governing VP, and others due to the fact that Past Tense can be used in contexts that involve reference to a future time, as is the case in example [5.12]. In this example, the Past Tensed VP governing the TE *Wednesday* induces a wrong value being assigned to this TE, value equivalent to the closest Wednesday before the reference time.

[5.12] *Authorities expected it to crest by* ***Wednesday*** *at the old trade town of Piacenza.*

This final model is further enriched with a module dealing with **the generic vs. specific problem**. In the TERN 2004 training data there are a total of

158 occurrences of the adverbs *today*, *yesterday* and *tomorrow*. As other authors have previously noticed, these instances are heavily skewed, in the sense that only 17 occurrences are generic, meaning that in 89.24% of the cases these adverbs have a specific usage (Ahn et al., 2005a). The classifier they train on the TERN data achieves an accuracy of 85%, and the authors give up the idea of trying to distinguish generic from specific usages, considering that better results are obtained by using only the baseline that considers all instances specific and in the case of the TERN data yields an accuracy of 89.24%. Mani and Wilson (2000) have also dealt with the ambiguity resulting from generic vs. specific meaning of TEs. They singled out the adverb *today*, which is most subject to this ambiguity, and developed a classifier which achieved an accuracy of 80% in the disambiguation of *today*. The module presented above in Section 5.2.6 performs slightly better compared to the baseline and the classifiers reported by other authors, with an accuracy of 92.40%. It manages to yield a small improvement in the normalisation results and the best F-score achieved so far for the normalisation task - 88%.

To contextualise these results, Table 5.3 contrasts the scores achieved by the full TE identification and normalisation system including the best normalisation model against results of the systems evaluated in the TERN 2004 competition. It is worth mentioning that the results of the systems that participated in TERN 2004 are evaluated on a smaller dataset than the TERN training data the present system was evaluated on.

	TIMEX2	ANCHOR_DIR	ANCHOR_VAL	MOD	SET	TEXT	VAL
TERN System-A	51.3%	0.0%	0.0%	9.4%	40.0%	26.1%	67.3%
TERN System-B	83.2%	53.9%	53.9%	66.7%	0.0%	77.6%	84.5%
TERN System-C	92.6%	76.0%	72.6%	77.4%	68.8%	83.9%	87.2%
TERN System-D	95.0%	61.7%	70.0%	10.5%	86.4%	84.9%	85.1%
TERN System-M	86.2%	66.9%	57.5%	17.8%	60.0%	62.1%	69.8%
TERN System-F	69.6%	45.7%	2.2%	0.0%	0.0%	58.4%	71.0%
Present System	*95.3%*	*76.1%*	*67.6%*	*84.3%*	*88.2%*	*86.3%*	*88.0%*

Table 5.3: The results of the systems evaluated at TERN 2004 against the results of the current system

Attribute	Annotator 1	Annotator 2	Annotator 3
TIMEX2	97.3%	97.2%	91.5%
ANCHOR_DIR	98.2%	87.9%	77.7%
ANCHOR_VAL	94.2%	85.6%	72.8%
MOD	98.3%	80.0%	56.4%
SET	98.0%	83.5%	83.3%
TEXT	96.3%	91.1%	89.4%
VAL	98.1%	93.9%	94.0%

Table 5.4: Official human annotator scores calculated against the final adjudicated TERN 2004 gold standard

The reason for evaluating the present system on the TERN 2004 training data is the unavailability of the test data due to copyright issues. Table 5.3 preserves the anonymity of the systems that participated in TERN 2004 and includes on the last line the complete results of the system presented in this chapter.

The official inter-annotator agreement figures released by the TERN 2004 organisers presented in Table 5.4 can provide a better picture of the difficulty of the task at hand. Three independent annotators have annotated the data, and the numbers in table 5.4 show each individual annotator's score when compared to the final adjudicated data [5].

This section presented the evaluation results obtained by implementing all the normalisation models described in Section 5.2. As a result of this evaluation, Norm-Local was found to be the best performing model for the normalisation of TEs. It was further enhanced with two modules performing direction disambiguation and generic vs. specific classification, thus achieving an accuracy of 88% in assigning a correct value to the TEs identified in text, a result which

5. The three annotators judged and reconciled the annotation cases they disagreed on, and a final gold standard was produced. No further information is available as to how this process was done.

is only 6% away from human performance for the same task. This final TE normaliser is further adapted to perform TIMEX3 normalisation, and the changes involved in this process are detailed in the following section.

5.4 Adapting the system for TIMEX3-compliant TE normalisation

5.4.1 The adaptation process

Section 4.5 presented the changes made in the TE identification system to comply with the TIMEX3 annotation guidelines described in 3.2.4. Changes are also required at the normalisation stage, and their number is much higher than those involved in the identification process.

Having seen how the differences between the two annotation schemes in terms of TE extent were tackled, the next step is adapting the information gathered during TIMEX2 annotation to fill the attributes corresponding to the TIMEX3 tag. The TIMEX3 attributes are: **tid**, **type**, **value**, **mod**, **temporalFunction**, **anchorTimeID**, **functionInDocument**, **beginPoint**, **endPoint**, **quant** and **freq**. Their usage is summarised in Table 5.5, and the methodology employed in filling their values is detailed below.

The attribute **tid** is automatically assigned so that each newly created TIMEX3 tag is assigned an unique ID number.

The attribute **type** is filled by looking at the class assigned throughout the TIMEX2 normalisation process according to the classification presented in Section 4.2. Most calendar point expressions are assigned the type DATE or TIME according to their granularity. Any calendar point expression at a

Attribute	Function	Example
tid	The unique ID number associated to each TIMEX3 expression.	tid="t0"
type	The type of the temporal expression: DATE, TIME, DURATION or SET.	type="DURATION"
value	The normalised form of the expression equivalent to TIMEX2 VAL.	value="P2D"
mod	The temporal modifiers also captured by TIMEX2 MOD.	mod="APPROX"
temporalFunction	Boolean attribute indicating that the value of the TE was determined via evaluation of a temporal function.	temporalFunction="true"
anchorTimeID	The ID of the temporal anchor used in evaluating the temporal function.	anchorTimeID="t1"
functionInDocument	The function of the TE in the document.	functionInDocument="NONE"
beginPoint	The ID of the TE representing the starting point of an anchored duration.	beginPoint="t1"
endPoint	The ID of the end point of an anchored duration.	endPoint="t2"
quant	The literal from the text that quantifies over a set-denoting TE.	quant="EVERY"
freq	The frequency at which the TE regularly reoccurs.	freq="3D"

Table 5.5: TIMEX3 attributes and their usage

granularity lower than the day-level is of type TIME, all other coarser-grained expressions are of type DATE. The only calendar points that are not assigned the type DATE or TIME belong to the class of embedded TEs. The annotation of the expression *two days before yesterday* presented in Section 4.5 shows not only the change of extent, but also a variation in the expression type when passing from TIMEX2 to TIMEX3 annotation. Such temporal expressions in an anchoring relation are called **anchored durations**. Even if the entire expression *two days before yesterday* denotes a DATE, the TIMEX3 standard specifies that the

two sub-expressions *two days* and *yesterday* should be annotated individually, thus assigning the type DURATION to the expression *two days*. The type DURATION is also assigned to all expressions denoting durations included in the class DURATION presented in Section 4.2. All expressions of frequency (class FREQUENCY) are assigned the type SET. TEs from the classes TOKEN and UNANCHORABLE are assigned the type DATE.

The attribute **value** is assigned the value of the TIMEX2 VAL attribute, except for certain embedded TEs and for set-denoting TEs. The embedded TEs that receive a different value are the anchored durations presented above. Their TIMEX3 value should reflect their new DURATION type, so their value is adjusted accordingly to a PXU-formatted value (X being the number of units of type U denoted by the expression). Another change appears in the case of set-denoting TEs, their value now changing from a value filled only with Xs to a value similar to the deprecated TIMEX2 PERIODICITY attribute. For example, the expression *every day* was given according to TIMEX2 the value "XXXX-XX-XX", while according to TIMEX3 it receives a value that is similar in formatting to the duration-type values, "P1D".

The attribute **mod** is directly inherited from the TIMEX2 mod attribute, receiving the same value.

The attribute **temporalFunction** is assigned the value *true* for every calendar point whose final value is calculated using a temporal anchor (i.e. all under-specified CALPOINT TEs). The same value is assigned in the case of TOKEN and DURATION expressions that previously required a temporal anchor to fill their ANCHOR_VAL attribute. For all other TEs, the attribute **temporalFunction** is assigned the default value *false*.

The value of the **anchorTimeID** attribute is the ID (i.e. the value of the tid

attribute) of the temporal anchor used in calculating the final value assigned to the VAL and ANCHOR_VAL attributes during the TIMEX2 annotation process.

The attribute **functionInDocument** is intended to capture the major milestones in the life of a textual document, such as the time the text is created, the time it is modified, published, released, received by the reader, or the time the text expires. However, in practice, only one value is used in the TimeML annotation applied to the TimeBank corpus. This value is CREATION_TIME and corresponds to the time the text is created, i.e. the Document Creation Time. All the other TEs present in text receive the default value "NONE" for this attribute. This attribute is automatically filled by the system using one simple rule. The first fully specified TE present in text with a granularity finer-grained or equal to the day-level is assigned the value "CREATION_TIME", all other expressions receive the value "NONE". This rule is used because the corpus possesses this characteristic and because the system has no access to the metadata included in the news articles. All metadata was eliminated from the TimeBank corpus to use plain text as input for the syntactic parser and for the present system.

The following four attributes are used to strengthen the annotation of durations and sets in TimeML.

The attributes **beginPoint** and **endPoint** are used for durations anchored by one or two TEs indicating their begin and/or end points. These attributes are filled with the IDs of the expressions serving as anchors. If only one of these points is made explicit in text, an empty TIMEX3 tag should be created to represent the missing point.

The attributes **quant** and **freq** should only be used when the expression is of type SET. The attribute **quant** captures the textual quantifier present in

the expression, and receives the value *EVERY* if the word *every* is part of the expression, or *EACH* if the word *each* quantifies the TE. The attribute **freq** contains an integer value and a time granularity that represent the frequency at which the temporal expression regularly reoccurs. This attribute is filled only for repetitive named expressions that are part of temporal cycles, such as *every Monday* or *every October 10*. The value assigned is the size of the cycle representing the period of time between two repetitions of the named expression (i.e. freq="1W" for *every Monday*, or freq="1Y" for *every October 10*).

5.4.2 Results and error analysis

The results obtained after evaluating the adapted TIMEX3 annotator (including the TE identifier described in Section 4.5 and the TE normaliser presented above) on the TimeBank 1.2 corpus, the reference resource annotated in compliance with the TimeML standard, are illustrated in Table 5.6. The evaluation is performed using the same scoring script employed by the TERN 2004 evaluation exercise, slightly modified by the author of the present work to score the TIMEX3 specific attributes instead of the TIMEX2 attributes the software was initially designed for. This script was chosen due to the high level of detail characterising its output.

The output of the scoring script is manually analysed to understand better the nature of the errors. A detailed error analysis for the system performing TIMEX3 annotation is presented below, with an emphasis on the **value** attribute, justifiable through its importance for other tasks relying on temporal expression annotation. The error analysis is guided by the error classes identified by the scoring script: **Incorrect**, **Missing** and **Spurious**.

	Possible	Actual	Correct	Incorrect	Missing	Spur	Precision	Recall	F-measure
TIMEX3	1414	1599	1383	0	31	216	86.5%	97.8%	91.8%
TEXT	1414	1599	1306	77	31	216	81.7%	92.4%	86.7%
type	1383	1381	1304	77	2	0	94.4%	94.3%	94.4%
value	1383	1372	1109	263	11	0	80.8%	80.2%	80.5%
mod	92	84	68	5	19	11	81.0%	73.9%	77.3%
temporalFunction	1383	1383	1325	58	0	0	95.8%	95.8%	95.8%
anchorTimeID	913	960	798	88	27	74	83.1%	87.4%	85.2%
functionInDocument	1383	1383	1383	0	0	0	100%	100%	100%
beginPoint	25	11	10	0	15	1	90.9%	40.0%	55.5%
endPoint	44	27	20	1	23	6	74.1%	45.5%	56.3%
quant	7	6	5	0	2	1	83.3%	71.4%	76.9%
freq	4	3	1	1	2	1	33.3%	25.0%	28.6%

Table 5.6: Evaluation results for the TIMEX3 annotator

Error analysis for the attribute value

- **Incorrect assignments:**

  The assignment of values to the TIMEX3 attributes is also an error-prone process. Most problems appear in the case of the attribute **value**, which is also the most important of all attributes characterising a TE. For this attribute, the scoring script detected 263 incorrect assignments.

  Nearly half of the errors (125) are errors made by the system in assigning the correct value.

  Among these system errors, a large number (52) appear in the case of expressions that make reference to financial quarters and financial years. They are mostly due to implementation errors revealed at the error analysis stage, such as not taking into account the fact that the normalised value of expressions that refer to the second, third and fourth financial quarters of a year should include in the year slot the previous calendar year to the one included in the anchoring TE (i.e. *the fourth quarter* uttered in an article dated January 1998 and referring to the fourth quarter of that particular financial year should be assigned a value of *1997-Q4*, and the system wrongly takes the year of the anchor TE and uses it to fill the year slot, yielding *1998-Q4*). These errors are easily rectifiable in the system implementation. Other errors made in the case of financial TEs include resolving wrongly expressions like *the latest quarter* or *the quarter* due to choosing the anchor wrong, and expressions like *the year-earlier quarter* due to implementation problems that appear when representing the unknown slots of the expression.

  There are also 39 errors made by the system when trying to solve the direction problem, meaning that in these cases the system fails to identify the

correct cycle for a named expression (e.g. for the expression *Friday*, the system assigns the value *"1990-08-17"* when the correct anchoring should have been in the previous week *"1990-08-10"*).

Eighteen errors are made when interpreting the meaning of a temporal expression, caused either by missing patterns or simply by errors made at the interpretation stage (e.g. the expression *Eight trading days* is wrongly assigned the value *"PXD"* when it should have received the value *"P8D"* because the pattern did not allow intervening words between the quantifier *eight* and the trigger word *days*, and as a consequence at the pattern matching stage only the word *days* is matched and mapped to a representation of *"PXD"*, and the words *Eight trading* are later included in the expression by the module that checks the TE's syntactic correctness).

Eight other system errors cover expressions headed by the word *period* that refer anaphorically to a certain period mentioned earlier in the text (e.g. the expression *the 1989 period* is assigned by the system the value *"1989"*, but this TE refers to a certain part of 1989 mentioned earlier in the text and its value should have been *"1989-Q3"*).

Six other system errors are failures of the system in locating the correct anchor, and two more errors are metaphorical usages of the expression *one day* with reference to the future (in the context *Farkas expressed the hope he* ***one day*** *follow in the footsteps of fellow astronaut John Glenn, who at 77 is about to go into space again.*), the system interpreting it literally as *"P1D"*.

Apart from system failures, a number of 111 human annotator errors have been identified in the cases marked as incorrect by the scoring script. Approximately half of these (50) are due to the fact that annotators have

assigned a value more fine grained than the granularity of the expression itself, while the system was developed in such a way that each expression received a value of the same granularity as that made explicit in the expression, following the recommendations of the TIMEX2 guidelines (for example the expression *last year* was assigned by the annotators the value *"1988-Q3"*, while the system filled in the value *"1988"*). A large number of human annotator errors (27) was noticed in the case of expressions having the granularity at week level, possibly due to the difficulty encountered by annotators to calculate week-level values (e.g. the expression *this week* was assigned by the system the value *"1998-W09"*, while the human annotator specified the value *"1998-WXX"*). Similar errors appear in the case of expressions denoting financial quarters. Errors have also been noticed in the annotation of expressions including timezone references (e.g. for the expression *08-15-90 1337EDT*, the human annotator assigns the value *"1990-08-15T13:37"* and forgets to add the ending that makes explicit the timezone: *"1990-08-15T13:37-04"*, that is specified in the annotation guidelines). It is easy to notice that certain errors are just human mistakes that clearly were made due to tiredness or lack of attention to detail (e.g. *July last year* is assigned the value *"1997-06"*, when the correct value should be *"1997-07"*).

Not all the cases marked as incorrect by the scoring script are due to system or human errors. There are 27 cases that cannot be considered errors due to the fact that sometimes the same value can be expressed in different ways, and both assignments are correct despite the fact that the textual representations differ. For example, given the expression *two thousand years*, the human annotator assigns to it the value *"P2L"*, while the system labels it as *"P2000Y"*, both values being correct.

- **Missing values**

  A number of 11 missing values were revealed. They correspond to fairly ambiguous references to time and include expressions such as: *a fairly lengthy period*, *the latest period*, *the corresponding period*, *the near term*, etc. The system was unable to assign any values to these expressions, and, considering that the annotators inserted values from the context of each expression, these cases were considered TEs missing assigned values.

- **Spurious cases**

  In the case of the **value** attribute, there were no cases of assignments made by the system not having a value assigned by the human annotators.

Error analysis for the attribute type

In the case of the attribute **type**, there are 77 cases where the values assigned by the system do not correspond to the values assigned by the annotators. An investigation of these cases reveals 42 errors made by the human annotators in assigning a type to a TE. Of these errors, 15 are cases of expressions referring to financial quarters or years that are incorrectly labelled as either being of type *TIME* or *DURATION*, when the guidelines indicate that expressions of granularity higher than times of day are of type *DATE*. One could argue that expressions which refer to financial quarters or years could be seen as durations, but the fact that they are semantically similar to seasons and even individual months justifies the approach taken when implementing the system that considers all these expressions of type *DATE* (e.g. *June* is considered to be of type *DATE* even if it spans 30 days, therefore *third quarter* is seen as being of the same type, the only difference being that financial quarters span several months). Another

11 cases are expressions that indicate indexicals formed using a duration followed by the post-modifiers *earlier*, *later* or *ago* (e.g. *a day earlier*). Such cases are annotated in the corpus as *DURATIONs*, but their semantics suggests a *DATE*, therefore the correct annotation for such cases should have been *DATE*. The other annotation errors are simply due to annotator negligence (e.g. *05/01/1998 09:13:00* is annotated as having the type *DATE*, when in fact the type is *TIME*).

32 system errors are due to the ambiguity of certain expressions that can express both *DATE* and *DURATION* (e.g. *the fiscal year* was assigned by the system the type *DATE*, when in fact it was annotated as a *DURATION* in the context of [5.13]), or expressions that can express both *DURATION* and *SETs* of times (e.g. in [5.14] the expression *a week* was considered by the system of type *SET*, when in fact it was used with a *DURATION* sense). Errors are also made by the system in the case of generic references to the past, present or future that are all automatically assigned the type *DATE*, when they are annotated as *DURATIONs* or *TIMEs* (e.g. *coming weeks*, *now*).

[5.13] *Mr. McNealy said the issues that hurt Sun's performance earlier this year are now "largely" behind the firm, and he indicated that Sun's profitability should increase throughout* ***the fiscal year****.*

[5.14] *After cabling world leaders about his intention to give Saddam Hussein a final deadline to exit Kuwait, he offered him* ***a week*** *to withdraw fully, instead of the four days he originally considered, because of objections from some European partners that four days seemed punitive and unrealistic.*

There are also cases of expressions that are assigned a correct type by both the human annotators and by the system, but the types differ due to the fact that human annotators annotate only a part of the expression identified by the system (e.g. the expression *a few days later* is assigned by the system the type *DATE*,

but in the corpus only *a few days* is marked as a TE of type *DURATION*; in such cases both types are considered correct for the textual chunks they are assigned to). However, since the scoring script counts automatically the cases where the values assigned by the human annotator are different from those assigned by the system, these cases are included in the class of incorrect assignments.

Error analysis for the attribute mod

- **Incorrect values**

  For the attribute **mod**, there are only 5 cases of incorrect assignments, most of them (4) being errors of the system in assigning a wrong value whenever an expression is modified by *nearly*. The value assigned is *EQUAL_OR_LESS*, when the guidelines specify that the value to be assigned should be either *LESS_THAN* or *APPROX*. These wrong assignments are due to a wrong value being correspondent to *nearly* in the lexicon.

- **Missing values**

  Most problems in the case of the **mod** attribute are due to missing values (19 cases), i.e. there are TEs that have a value annotated in the corpus for the attribute **mod**, but the system fills in no value. Some of these cases are ambiguous semantically and it is hard to tell whether they really required the **mod** attribute to be filled in (e.g. for the TE *a fairly lengthy period* the annotator considered that the **mod** attribute should receive the value *"EQUAL_OR_MORE"*). In other cases it is very clear that no value should have been assigned to **mod** (e.g. *the past two months* was assigned the value *"BEFORE"*). The remaining cases are system errors at the identification stage, and due to these errors the modifier is not identified, no label being therefore

assigned (e.g. in the case of the TE *1990 and beyond*, only *1990* is annotated by the system, and since the post-modifier is not identified, its semantics is not captured in the value of the **mod** attribute).

- **Spurious values**

  The 11 spurious cases present in the case of **mod** are mainly due to the fact that the modifiers of the TEs involved are not annotated in the gold corpus, and this is why no value was assigned by the human annotators to **mod** (e.g. *earlier this month* is fully annotated by the system that assigns the value *"START"* to **mod**, but in the gold corpus only *this month* is marked up, and no value is assigned to **mod**). There is only one case where the modifier is included in the TE in the manual annotation, but no value is assigned to **mod** (*the end of the month* - the system assigns the value *"END"* to **mod**).

Error analysis for the attribute temporalFunction

In the case of the attribute **temporalFunction**, there are 58 cases marked as incorrect assignments by the automatic scorer. A close look at these cases shows that 31 of them are again due to errors in the human annotation, mostly in the case of durations that appear in contexts which might make someone believe that functions could be used for temporal calculations (see [5.15]). The problem here is the mis-interpretation of the guidelines, as the guidelines indicate that the value *"true"* for **temporalFunction** should be assigned only when the value of the TE was determined via evaluation of a temporal function. In these cases no function is used when assigning a value to the expression which is of type duration, but a function could be used at a later stage when the temporal reasoner would interpret the TE contextualised by the signal *in*.

[5.15] *The stock would be redeemed in* ***five years*** *[...].*

The other errors are simply caused by annotator negligence (e.g. for the TE *03/08/1998 06:26:00*, the annotators fill in the value *"true"* for the **temporalFunction**, when obviously no function is used in assigning a value to this expression as it is fully specified).

System errors account for 25 cases of incorrect **temporalFunction** assignments. Many errors (9) apply to expressions headed by the word *period* (e.g. *the latest period*) which the system does not attempt to resolve and automatically assigns them the value *"false"* for **temporalFunction**, when the correct value would have been *"true"* as they anaphorically refer to another TE in the previous context. Other errors are due to problems that appear at the identification stage that prevent one from identifying the full meaning of an expression (e.g. in the case of the TE *the next 12 to 18 months*, the pattern that identifies this expression is a simple duration pattern that is able to pinpoint the expression *18 months*, the rest of the expression being identified by the syntactic correctness checker, thus preventing the proper interpretation for this TE: the normalisation module sees it as a simple duration, and not as an anchor duration as it would see *the next 18 months*). Errors also appear due to metaphorical usages of expressions like *one day* that refer to the future, but the system sees them as simple durations without any need for a temporal function.

There are also two cases where the system and the human annotator do not agree on the extent of the TE they annotated, but for which the value of the **temporalFunction** attribute is correct with respect to that extent. The human annotator and the system assign two different **temporalFunction** values to two different extents of the same expression (e.g. *three days later* is assigned the value *"true"* by the system, while the annotators mark up the extent *three*

days with the value *"false"*). Both values are correct when viewed from each annotator's perspective, but according to the scoring script the system assignment is considered incorrect.

Error analysis for the attribute quant

- **Missing values**

  The two cases of missing values for the attribute **quant** are human annotation errors. This attribute should receive as value the literal from the text that quantifies over a set-denoting TE, and for these two cases (*a year*, *fourth quarters*) there is no explicit mentioned literal like *every* or *each* quantifying them.

- **Spurious values**

  There is also one value for **quant** filled by the system correctly, but due to the fact that the annotators did not annotate the expression *each July*, there is obviously no value assigned to this attribute.

Error analysis for the attribute freq

- **Incorrect values**

  There is one case marked as incorrect assignment for the attribute **freq**, and the expression to which it applies is *Tuesday nights*. It is not a true error, but just a difference in the representation of the value: the human annotators label the frequency as *"7D"* while the system assigns it *"1W"*, these values being equivalent as *seven days* equals *one week*.

- **Missing values**

Two cases of missing values for the **freq** attribute are identified. Both are system errors due to the system not identifying that *Monday* and *the past three summers* are set denoting expressions.

- **Spurious cases**

  One value of the **freq** attribute annotated by the system does not find its correspondent in the gold standard, and this case applies to the same expression that was also missing from the human annotation of the **quant** attribute: *each July.*

Other attributes

An error analysis for the attributes **anchorTimeID**, **beginPoint** and **endPoint** is not presented in this thesis, due to the difficulty of investigating each case given that the gold standard and the system annotation use different sets of IDs for the TEs they identify, and each error case would involve tracking all the IDs involved in finding the anchor that contributed to a certain value being assigned to those affected TEs. In the future, a module will be implemented to help in presenting the errors so that tracking how these attributes were assigned would be more user-friendly and would allow a clear investigation.

This section focused on presenting the changes made to adapt the TIMEX2 normaliser to perform TIMEX3-compliant normalisation, on evaluating the resulted TIMEX3 annotator, and on analysing the errors that appeared throughout this process. The good results obtained in the evaluation process have shown that the automatic TIMEX3 annotation can be done with a high reliability. Unfortunately it is not possible to compare the results obtained by the present system with previous efforts made towards TIMEX3 annotation. This

is due to the fact that no prior appropriate evaluation of systems that perform TIMEX3 annotation has been made. The only system that performs TIMEX3 annotation is GU-Time (Mani and Wilson, 2000), but no evaluation results for TIMEX3 annotation have been reported. GU-Time is only benchmarked on the TERN 2004 data, reporting only figures for TE extent mark-up and for assigning a value to the attribute VAL. The F-measure figures reported by GU-Time on the TERN 2004 data are: 85% for TIMEX2 partial matching of the TE extent, 78% for TEXT full matching, and 82% in assigning a value to VAL [6]. However, the GU-Time results on the TERN 2004 training data can be compared with the results obtained by this system for TIMEX2 annotation.

This section also included a detailed discussion of the errors and problems encountered during the evaluation process. This discussion reveals that many problems are due to inconsistencies in the annotation of the TimeBank corpus, an observation that is also confirmed by other researchers (Boguraev and Ando, 2005, 2006). The detailed analysis of the TimeBank TIMEX3 annotation included in this section is one of the major contributions of this chapter. It shows that TimeBank, the most important corpus available for studying various temporal phenomena, still requires effort invested in ensuring high annotation quality and consistency. But for this effort to be well invested, the TimeML guidelines should be revised and improved with detailed and straightforward information about how each type of TIMEX3 expression should be annotated.

6. These figures can be found in Inderjeet Mani's tutorial on *Temporal Information Extraction from Natural Language* held as part of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07). His presentation is available online at: http://www.cs.mu.oz.au/research/lt/nlp06/materials/Mani/temporal-tutorial.ppt

Better guidelines would greatly help human annotators, would lead to an improvement in inter-annotator agreement and to the development of a reliable resource for TIMEX3 and TimeML annotation, but would also prove invaluable for automatic annotation system developers.

5.5 Conclusions

This chapter investigated the normalisation stage of the TE annotation process. The normalisation process identified the value designated by a given temporal expression, value captured using different attributes specified by a chosen annotation scheme. It relied on the information gathered at the TE identification stage that was described in Chapter 4.

During normalisation, the values of the attributes were either extracted from the expression itself, or calculated using the attribute values of another TE which served as anchor time. An important problem in normalisation is choosing the anchor that contributes to finding the value associated with an under-specified temporal expression. Several normalisation models were proposed, each one of them following a different methodology for choosing the temporal anchor for under-specified expressions. Section 5.2 focused on the methodology used for normalisation, and discussed several normalisation models. The comparative evaluation of these different alternative models for TE normalisation was captured in Section 5.3. For evaluation purposes, the TIMEX2 annotation scheme was initially adopted, due to its high level of detail and complexity among existing schemes. The system is able to identify the correct temporal value associated with a TE in 88% of the cases, which is a very good result considering that human performance for this task is about 94% (see Table 5.4 for more details).

The TE annotator developed was then adapted to the TIMEX3 annotation scheme, part of the TimeML standard, and another evaluation was performed on the TimeBank corpus. The changes involved in the normalisation process, as well as the results obtained by evaluating the TIMEX3-adapted annotator were described in Section 5.4. The evaluation results show that the system is able to identify the correct value associated with a TIMEX3 expression in 80.5% of the cases. Section 5.4 also featured a detailed error analysis that identified the main problems that appeared in the TIMEX3 annotation process.

The TIMEX2 and TIMEX3 evaluations and the corresponding error analysis have shown that the system developed as part of this work is a reliable tool for the annotation of temporal expressions that can be easily adapted to a new annotation standard. They have also shown that this system can be used for the cross-validation of annotated data. The system has been initially developed to comply with the TIMEX2 annotation guidelines, but the architecture and representation were designed to be as general as possible. The fact that it has been developed using well-documented and reliable guidelines, evaluated on a sound annotated corpus yielding very good results, and then adapted to the TIMEX3 standard, has allowed it to identify several problematic issues concerning the annotation of the TimeBank corpus. A manual analysis of the differences between the TIMEX3 annotation made by humans and the one made by the system revealed a very high number of errors present in the human annotation that justify the system's drop in performance when adapted from the TIMEX2 to the TIMEX3 annotation standard.

Chapter 6

Events

6.1 Overview

This chapter describes the methodology adopted in this work for the annotation of events expressed by means of finite and non-finite verbs in natural language texts. Events are most of the time expressed using verbs, but certain nouns and adjectives are also capable of denoting events. An important step towards identifying and annotating events in text consists of being able to deal with verbal events. This research aims to identify a general method to be employed in the identification and annotation of verbal events according to the TimeML annotation standard.

One important problem posed by the event annotation process is the ability to classify verbs as belonging to a certain event class. In this thesis, the classification problem is solved by carrying out an annotation process on all the verbs present in WordNet 2.0 (Fellbaum, 1998), and assigning to each verb its most relevant event class - that is the event class that covers most of that verb's meanings. This approach is similar to the one often adopted for the Word Sense Disambiguation task, where the most frequent sense is assigned to every occurrence of a word.

Section 6.2 describes events and the problem of event classification from the

perspective of this research. It also identifies drawbacks of existing approaches for event identification and classification, and finally distinguishes the event classes to be employed in the annotation process carried out in this work.

Section 6.3 reports on the annotation process that results in each English verb being assigned an event class. This annotation is done according to the classification decided in Section 6.2, a classification that will allow locating events in time and aid in obtaining the temporal interpretation of a given natural language text. This section also features a detailed discussion of the issues raised throughout the annotation process, as well as of the cases where the annotators disagree, at the same time measuring inter-annotator agreement.

The resulting annotated resource is extremely important for an automatic system that aims to mark up events according to the TimeML standard. The functionality of such a system involves first identifying verbal events in a text, and then annotating them with TimeML-compliant information. Section 6.4 describes the methodology adopted in this work for identifying events expressed using verbs. Section 6.5 presents the annotation process of the events identified at the previous stage, a process that relies on the resource associating each verb with an event class. This method is evaluated on TimeBank (Pustejovsky et al., 2006), the reference corpus annotated with this type of temporal information.

In the last section, conclusions are drawn and future directions of research in the area of event annotation considered.

6.2 Events and their classification

This research relies upon the TimeML specification language (Pustejovsky et al., 2003; Saurí et al., 2006), which has been adopted worldwide as the inter-lingua

for temporal markup, and on the TimeBank corpus, the proof of concept for the TimeML specifications. TimeML considers **events** as "a cover term for situations that *happen* or *occur*" (Pustejovsky et al., 2003). The TimeML specifications also consider as events "those predicates describing *states* or *circumstances* in which something obtains or holds true" (Pustejovsky et al., 2003).

Events may be expressed by means of tensed or untensed verbs, nouns, adjectives, predicative clauses, or prepositional phrases, but, for a simplification of the annotation process TimeML has imposed certain rules in order to select the word or group of words to be annotated as events by applying the test of headedness, i.e. only the head word of the group that denotes an event should be annotated. By looking only at the words annotated according to these rules, statistics extracted from the TimeBank corpus reveal that the annotated extent of an event is in 64.5% of the times a verb, in 28% of the cases a noun, 3.4% of events are adjectives, 0.3% prepositions, while 3.8% are assigned a part-of-speech category called OTHER (these events are, in most cases, numeric expressions or adverbs).

Since verbal events are most frequently encountered in text, the present study focuses on the identification and classification of events expressed by means of verbs, also aiming in the future to identify a suitable methodology for events expressed using nouns.

To identify events, one existing approach is to consider all verbs events, with the exception of the verb *to be* and of several other forms of generics (Harabagiu and Bejan, 2006), while another approach, besides this restricted set of verbs, also considers as events certain nouns and the adjectives annotated as such in TimeBank (Saurí et al., 2005). Other domain independent approaches consider as an event a text unit, at a coarser-grained scale the sentence (Hitzeman et al.,

1995), and at a finer-grained scale the clause (Mani and Shiffman, 2005).

The method employed until recently by other researchers in classifying events into event classes was very preliminary, and involved tagging events with the class that was most frequently assigned to them in TimeBank (Saurí et al., 2005). Newer approaches described in detail in Section 3.5 train different types of classifiers to distinguish between event classes, mainly by looking at co-occurrence patterns that manifest resemblance to events annotated in TimeBank. The first method offers good performance only for words annotated in TimeBank and it obviously does not cater for verbs, nouns or adjectives not included in TimeBank, while the second method tries to overcome TimeBank's limits but is still highly dependent on TimeBank and the performance is around 60%, thus not ensuring reliability. This is where this research and the methodology proposed below bring a novel contribution by offering the research community a reliable method to identify and classify verbal events in any natural language text, irrespective of their appearance in TimeBank. Unlike all previous methods, this method does not depend on the information annotated in TimeBank, information that can be sometimes unreliable due to multiple annotation inconsistencies.

The main idea of the present work is to annotate each English verb present in WordNet with an event class, an annotation which aims not only to be useful to the research community in the assignment of an event class to a given verb, but also to be a starting point that can afterwards be refined at verb sense level, or transferred to other languages using the WordNet ILI (Inter-Lingual Index) alignment. The annotation efforts presented below are also captured in Puşcaşu and Barbu-Mititelu (2008).

Since the target of this research is to obtain a tool capable of annotating any text with TimeML compliant temporal information, the event classes defined by

TimeML were considered as the starting point in this investigation. These classes are:

- **REPORTING**: these events describe the action of a person or an organisation declaring, narrating or informing about an event, so their function is to associate the source of information with the reported event (example [6.1]);

  [6.1] *John **said** he bought some wine.*

- **PERCEPTION**: this class includes events involving the physical perception of another event (example [6.2]);

  [6.2] *Mary **saw** John carrying only beer.*

- **ASPECTUAL**: these events capture the aspectual predication on different facets of another event's history: initiation, reinitiation, termination, culmination, continuation (example [6.3]);

  [6.3] *The search party **stopped** looking for the survivors.*

- **I_ACTION**: an intensional action event introduces an event argument describing an action or situation from which we can infer something given its relation with the I_ACTION event (example [6.4]);

  [6.4] *Bill **attempted** to save her.*

- **I_STATE**: this class contains states that refer to alternative or possible worlds (example [6.5]);

  [6.5] *Bill **wants** to teach on Monday.*

- **STATE**: a circumstance in which something holds true (example [6.6]);

  [6.6] *They **lived** in Netherlands for 2 years.*

- **OCCURRENCE**: an occurrence event is defined as something that happens or occurs in the world (example [6.7]).

  [6.7] *John* ***drove*** *to Boston.*

An analysis of these event classes and of TimeBank, the corpus annotated with TimeML compliant temporal information, reveals many annotation inconsistencies, in the sense that the same verb in very similar contexts is annotated with different classes (for example the verb *launch* in the context *launch the offer* is in one case annotated with the class OCCURRENCE, and in the other case with I_ACTION). Even the official inter-annotator agreement figures for TimeBank (see Section 3.3.2) reveal many inconsistencies, the inter-annotator kappa agreement for the event class being 0.67. This figure also illustrates the fact that event annotation is not a trivial task, even for humans. The classes OCCURRENCE and I_ACTION both include situations that happen, occur or involve change, the only difference between them being the fact that the I_ACTION event has an event argument and provides factuality information about its argument, while the OCCURRENCE event does not. The same applies to the classes I_STATE and STATE. The events included in the classes I_ACTION and I_STATE capture the factuality of their argument event. Since the investigation of event factuality is a complex topic on its own, considering that it alone represented the focus of a PhD thesis (Saurí, 2008), it has been considered wise to leave the factuality problem aside and to establish more achievable goals. To make the human annotation process easier, and the targeted automatic annotation process feasible for an automatic tool, the present work focuses on a reduced set of event classes obtained by merging the OCCURRENCE and I_ACTION classes into only one class (OCCURRENCE), and by also merging

the STATE and I_STATE classes into one class (STATE), thus obtaining the following simplified set of event classes:

- **REPORTING**: corresponds to the TimeML REPORTING class ([6.1]);
- **PERCEPTION**: is the same as the TimeML PERCEPTION class ([6.2]);
- **ASPECTUAL**: corresponds to the TimeML ASPECTUAL class ([6.3]);
- **OCCURRENCE**: covers the TimeML OCCURRENCE and I_ACTION classes ([6.7] and [6.4]);
- **STATE**: includes the TimeML STATE and I_STATE classes ([6.5] and [6.6]).

Even if there are reasons to differentiate the OCCURRENCE and I_ACTION, as well as the STATE and I_STATE events pragmatically, this research will place higher relevance on the resemblances which bring these classes together, and will neglect the differences. Differentiating between these classes can be seen as a totally different task, one that has already been dealt with in great detail in Saurí's PhD thesis (2008). In contrast to Sauri's work, which focuses on the problem of event factuality that captures the differences between the OCCURRENCE/STATE and I_ACTION/I_STATE classes, this research differentiates between the 5 classes enumerated above (REPORTING, PERCEPTION, ASPECTUAL, OCCURRENCE, STATE).

Each of the five classes in the reduced set has different temporal properties. For example, a REPORTING event most commonly happens after the reported event, while perceived events happen roughly at the same time as the PERCEPTION events. The temporal consequence of ASPECTUAL events is that they indicate different stages of their argument event (beginning, end, continuation). OCCURRENCE events cover situations that involve change,

processes consisting of different stages, or situations that have duration and involve an end result. STATE events cover situations that do not involve change over time. In the case of two consecutive events, typically an OCCURRENCE takes place just after a preceding OCCURRENCE, while a STATE overlaps a preceding OCCURRENCE.

The event classes presented above will represent the focus of the annotation process that is described in the following section.

6.3 Annotation of WordNet verbs with TimeML classes

The annotation process takes place in two stages, at the first stage each verb is assigned one WordNet lexicographic file, while at the second stage each verb in turn is assigned one event class by two independent annotators.

6.3.1 Mapping verbs to WordNet lexicographic files

WordNet verb senses are grouped into 15 lexicographic files:

- verb.body
- verb.change
- verb.cognition
- verb.communication
- verb.competition
- verb.consumption
- verb.contact

- verb.creation
- verb.emotion
- verb.motion
- verb.perception
- verb.possession
- verb.social
- verb.stative
- verb.weather

Lexicographic files were developed by lexicographers following a complex relational analysis of lexical semantics. Each file includes synsets (lists of synonymous word senses that are interchangeable in some context) belonging to the same syntactic category and relations that hold between synsets (e.g. hypernymy, hyponymy, antonymy, etc.). This research looks at the 15 lexicographic files corresponding to verbs, and relies on the assumption that verb senses included in one lexicographic files would manifest a preference for certain event classes: for example one would expect that verb senses included in the **verb.communication** file would be typically classed as either REPORTING or OCCURRENCE events.

Since one verb can have more senses, there are cases when not all verb senses are in the same lexicographic file. In fact, from a total number of 11,306 verbs present in WordNet 2.0, only 7,437 verbs have all their senses in the same lexicographic file, for the remaining 3,869 verbs the senses are scattered among several lexicographic files. The first stage in the annotation process is assigning to each English verb only one lexicographic file. The assigned file is the one that maximises the score:

$$\texttt{score(file}_i\texttt{)} = \sum \texttt{(1/j)}$$

for each j ranging from 1 to the number of senses of the analysed verb.

In the above formula, j is the sense number and $\texttt{file}_i$ is the corresponding lexicographic file assigned to sense j. This formula chooses the lexicographic file that covers most of the important senses of a verb (as one can notice, a higher sense number corresponding to a more frequent sense gives a higher score to its lexicographic file).

6.3.2 Annotation process

Annotation

After each verb was assigned one lexicographic file, two annotators examined each lexicographic file and assigned to each verb one of the five event classes described above (REPORTING, PERCEPTION, ASPECTUAL, OCCURRENCE, STATE). Both annotators approached the annotation from two different perspectives and employed different resources.

The first annotator looked only at those WordNet senses and corresponding synsets that motivated the verb's inclusion in the assigned lexicographic file and identified the event class that offers the highest coverage of those senses.

The second annotator looked up each verb in the *Oxford English Dictionary*, eliminated all obsolete and rare senses, and assigned the class that, according to the annotator's intuition, covered best the remaining senses.

One could argue that annotating verbs for their event type outside a context is not a proper way of doing it, as words do not have meaning in isolation, but only in the context of a sentence. However, this annotation is not done entirely outside a context, as the annotators have access to the lexicographic definition and they

use their intuition for how this word would be used in a context. Therefore, this work relies on the assumption that the core meaning of a word can be captured in a lexicographic definition, and the context only favours refinements of that meaning (with some semantic traits being blocked or, on the contrary, encouraged to manifest in certain word combinations).

At the end of the annotation process, the cases of agreement/disagreement were carefully analysed. This analysis revealed that, out of 11,306 verbs, the same class was assigned by both annotators in 10,945 cases, meaning an absolute agreement of 96.80%. By investigating the cases of disagreement, certain issues that were not clearly specified in the annotation guidelines were discovered. The next step was to clarify the guidelines and to revise the annotation accordingly.

Revision of the guidelines and of the annotation

The cases of disagreement revealed annotation errors due to issues in the guidelines that required further clarification.

One issue refers to events that were wrongly annotated as REPORTING. Certain communicative verbs were classified as REPORTING, even if they do not have the ability to report about other events they would take as arguments (in case they could have arguments). Here are some examples of verbs wrongly annotated as REPORTING: *counsel*, *talk*, *compliment*. These verbs cannot occur with arguments denoting events they talk about. One should also be aware that the annotator's choice was influenced by the verb semantics filtered through that person's idiolect and life experience. In the case of the verb *disagree* for example, it is well known that disagreement is most frequently expressed verbally, so, as a result, this verb was initially categorised by one annotator as REPORTING. The same misinterpretation was to blame for some verbs being initially annotated as

REPORTING, and only on second thought as OCCURRENCE: *decree*, *swear*, *badmouth*, etc.

Similarly, some verbs were wrongly annotated with the class PERCEPTION, when they lacked the ability to describe the physical perception of another event, even if they referred to physical perception. For example the verb *suffer* should have been annotated with the class STATE, while the verb *hurt* should have received the class OCCURRENCE.

Another issue was that, in order to annotate a verb as ASPECTUAL, that verb should, in its most frequent usages, take another event as argument, to whose aspectual facets it should refer. Since this was not clearly expressed in the annotation guidelines, verbs like *break_out* or *abrogate* were wrongly annotated as ASPECTUAL, even if both *break_out* and *abrogate*, with their most frequent senses, neither take other events as arguments, nor do they refer to a certain stage in an event's evolution.

Therefore, whenever deciding whether a certain verb is a REPORTING, PERCEPTION or ASPECTUAL event, the annotators were advised to imagine in which contexts that verb would typically be used in, and whether those contexts frequently involved that particular verb taking another event as argument.

An important problem observed by analysing the disagreement cases was that the boundary between what was defined as STATE and what was defined as OCCURRENCE was not clear-cut. In many such cases the verbs involved express inner or physiological processes, which one of the annotators initially considered STATEs, and the other OCCURRENCEs: *didder*, *retrospect*, *gestate*.

After discussing all the above mentioned issues and clarifying the guidelines, both annotators independently adjusted their annotations accordingly for the verbs they did not agree upon, each annotator reconsidering the class they would

assign to those verbs, without knowing the other annotator's decision. Finally, inter-annotator agreement was measured on the resulted annotations. Out of 11,306 verbs, the two annotators agreed on the same class being assigned to 11,087 verbs, yielding an absolute agreement of 98.06%. Cohen's kappa statistics (Cohen, 1960), which also takes into consideration the proportion of chance agreement, reveals a kappa score of 0.87, indicating a very high agreement.

Final Decision

The remaining cases of disagreement (accounting for 219 verbs) were then submitted to a third annotator, who was asked to assign to each verb one of the five event classes. A voting scheme was then applied to the three annotations, and each verb was assigned the class two out of three annotators agreed on. Still, there were 16 verbs for which the three annotators chose three different classes. For example, in the case of the verb *give_out*, one annotator chose the class REPORTING (as it has the meaning *to announce; proclaim; report*, see [6.8]), another annotator chose the class STATE (as it has the meaning *to emit*, see [6.9]), and the third annotator chose the class OCCURRENCE (as it has the meaning *to break down, get out of order, fail*, see [6.10]).

[6.8] *He* ***gave out*** *at Macao, that he was bound to Batavia.*

[6.9] *The gold* ***gave out*** *its red glow.*

[6.10] *The Ruby's engines* ***gave out*** *for a time.*

The final classes for these 16 verbs were decided by a fourth annotation.

At this point each WordNet verb had a unique class assigned to it, and the resulted resource was ready to be employed in a system capable of annotating verbal events. The development of this system is described in detail in the following sections.

6.4 Identification of verbal events in text

One of the goals of this research is to design a methodology for identifying and annotating verbal events in natural language texts. Two tasks are involved in achieving this goal: the first is concerned with the identification of events in the sense of discovering the textual extent of verbal events, and the second task requires filling in the values of the attributes that characterise an event according to the TimeML specifications. This section focuses on the first task, verbal event identification, and Section 6.5 on how the attribute values are assigned to each verbal event.

The identification of verbal events in natural language texts is achieved by first parsing the input data with Connexor's FDG parser (Tapanainen and Jarvinen, 1997), then analysing the output and identifying the verbs present in text. The experiments presented in this chapter are performed on the TimeBank 1.2 corpus, the reference corpus that includes events annotated according to TimeML. Since the verbal event identification system is designed to work on any natural language text, the TimeBank articles are first converted to plain text by eliminating all XML tags, and then processed using Connexor's FDG parser. This parser returns information on a word's part of speech, morphological lemma and its functional dependencies on surrounding words. This information is useful for the identification of verbal events, as well as for finding the values of most TimeML attributes.

On the basis of the information provided by the syntactic parser, the system then identifies finite verb phrases and non-finite verbal constructions with the aim of marking up their syntactic heads as events. The processing is done separately for finite and non-finite verbs due to several reasons. First of all, the grammatical

structure of the verbal groups headed by finite verbs is different from the one of non-finite constructions, so the system handles them differently. Another reason for processing them separately is the emphasis on finite verb events and on the syntactic structures they dominate (clauses and sentences) encountered in previous research concerning events (see Section 3.5 for more details). This leads to the hypothesis that finite verbs are more relevant than non-finite verbs in the context of event annotation, so it was interesting to see the differences between the two classes in this context. In the following, the process of event identification and its evaluation on TimeBank 1.2 is presented separately for finite and non-finite verbal events.

6.4.1 Identification of finite verb events

The information provided by Connexor's FDG parser is employed to detect the full extent of the finite verb phrases that appear in a text. The head of each identified verb phrase, which is usually the last word in the group, is then marked as an event, except in the case when the head is any form of the verb *to be*. This exclusion is due to the TimeML guidelines which clearly specify that any occurrence of the verb *to be* as finite main verb should not be labelled as event. Therefore, all finite main verbs except the verb *to be* are considered events.

To compare the performance of this purely syntactic finite verb event identifier against TimeBank, only the events annotated as finite verbs in TimeBank were considered. The criterium employed to select them was to extract those events for which the attribute **pos** had the value VERB, and the attribute **tense** had any of the values PAST, PRESENT, FUTURE or NONE. Even if in many cases non-finite verbs in the infinitive were annotated with the class NONE for the attribute **tense**, when this attribute should have received the value INFINITIVE, this was

considered to be an error in the TimeBank annotation, and no change was made in the way the finite verb events constituting the gold standard were extracted from TimeBank.

When comparing the finite verb events identified by the system with the ones annotated in TimeBank, the following figures are revealed:

- there are 3,845 finite verb events annotated in TimeBank.
- the system identifies 4,466 finite verb events in all TimeBank articles.
- in 3,602 cases the finite verbs identified by the system coincide with those annotated as finite verb events in TimeBank. This leads to a precision of 80.65%, a recall of 93.68%, and an overall f-measure of 86.68% in identifying finite verb events. The lower precision obtained in identifying finite verb events is largely due to the fact that no attempt is made to identify verbs with generic usages or verbs present in headlines in order to avoid their annotation.
- in 3,738 cases the finite verbs identified by the system are annotated as events in TimeBank. Of these, 3602 are annotated as finite verb events, 68 as non-finite verb events, 35 have the part of speech set on NOUN, 29 on ADJECTIVE, and 4 on OTHER. A close look at those finite verb events that appear annotated in TimeBank as either non-finite verbs, nouns or adjectives revealed 86 errors caused by the syntactic parser, and 46 cases wrongly annotated in TimeBank (18 finite verbs wrongly annotated as non-finite, 15 wrongly annotated as nouns, and 13 wrongly annotated as adjectives).

When compared to TimeBank, the system identifies 728 (4,466 - 3,738) more events than those annotated in TimeBank. An investigation of these cases shows that 284 verb occurrences should have been annotated in TimeBank and were

not. The remaining 444 finite verb events identified in excess are due to different reasons which are explained below.

There are 318 cases that should not receive an annotation according to the guidelines. Generic usages of verbs are not supposed to be annotated, and, since no attempt is done to identify generic usages of finite verbs, they are annotated in 140 cases (e.g. [6.11]). Events occurring in article headlines should not receive an annotation, and there are 88 cases of finite verb events that the system identifies in headlines (e.g. [6.12]). Modal verbs and auxiliary verbs not followed by a main verb are also excluded from annotation, and the system annotates such verbs in 83 cases (e.g. [6.13] and [6.14], respectively). There are also finite verbs appearing in fixed phrases that do not contribute to the meaning of the sentences and they should not be annotated (the system annotates 7 such finite verbs, e.g. [6.15]).

[6.11] *Ethnic Albanians* ***comprise*** *90 percent of the population in Kosovo, but Serbs maintain control through a large military and police presence.*

[6.12] *Saddam* ***Seeks*** *End To War With Iran.*

[6.13] *We will continue to do everything we* ***can*** *to establish what has happened.*

[6.14] *Service industries also showed solid job gains, as* ***did*** *manufacturers, two areas expected to be hardest hit when the effects of the Asian crisis hit the American economy.*

[6.15] *You* ***know****, since he's been here the stock skyrocketed so, yeah I think he's doing the right thing.*

There are 126 errors of identification produced by the syntactic parser. These comprise all those cases in which nouns (e.g. [6.16]), adjectives (e.g. [6.17]), adverbs (e.g. [6.18]), prepositions (e.g. [6.19]) or conjunctions (e.g. [6.20]) were

annotated as finite verbs, and also cases of ungrammatical sentences (e.g. [6.21]), and non-finite verbs (e.g. [6.22]) that are tagged as finite ones.

[6.16] *The Pentagon said that Defense Secretary Dick Cheney is considering urging Bush to order a national callup of armed forces* ***reserves*** *for active duty because of the drain on units sending soldiers abroad.*

[6.17] *Last year, Russian officials assailed Ukraine for holding* ***joint*** *naval exercises with NATO in the Black Sea an area Moscow considers its own turf.*

[6.18] ***Live*** *from Atlanta, good evening Lynne Russell, CNN headline news.*

[6.19] *His advisers said the results reflected not just from balancing the budget, but also initiatives* ***like*** *improved access to education and training and the opening of foreign markets to trade.*

[6.20] *Prime Minister Benjamin Netanyahu told his Cabinet on Sunday that Israel was willing to withdraw from southern Lebanon* ***provided*** *Israel's northern frontier could be secured.*

[6.21] *In Hong Kong, is always* ***belongs*** *to the seller's market.*

[6.22] *In a long verbal attack* ***read*** *on Iraqi television Thursday, Saddam repeatedly called Bush "a liar" and said a shooting war could produce body bags courtesy of Baghdad.*

6.4.2 Identification of non-finite verb events

In a similar manner to the above procedure followed for the identification of finite verb events, non-finite verb events are also detected on the basis of the output provided by Connexor's FDG parser. Using the lexico-syntactic information given by the parser, the full extent of all non-finite verb constructions is first identified. As in the case of finite verbs, only the head of each non-finite verb construction is automatically annotated as an event. The only exception to this process is any

non-finite form of the verb *to be*.

To compare the system output with the gold standard, the non-finite events annotated in the gold standard corpus (TimeBank) are extracted by selecting only those events for which the attribute **pos** has the value VERB, and the attribute **tense** ranges over the values INFINITIVE, PRESPART and PASTPART.

A comparison between the non-finite verbal events annotated in TimeBank and the ones automatically identified by the system revealed the following:

- there are 1,274 non-finite verb events annotated in TimeBank.
- the system identifies 1,819 non-finite verb events in all TimeBank articles.
- in 1,136 of the cases the non-finite verb events identified by the system are also annotated in TimeBank. This leads to a precision of 62.45%, a recall of 89.16%, and an overall f-measure of 73.45% in identifying non-finite verb events.
- in 1,356 cases the non-finite verbs identified by the system are annotated as events in TimeBank. Of these, 1,136 are annotated as non-finite verb events, 123 as finite verb events, 84 have the part of speech set on NOUN, 12 on ADJECTIVE, and 1 on OTHER. A careful examination of those non-finite verb events that appear annotated in TimeBank as either finite verbs, nouns or adjectives revealed 125 cases wrongly annotated in TimeBank (70 non-finite verbs wrongly annotated as finite, 48 wrongly annotated as nouns, and 7 wrongly annotated as adjectives), as well as 94 parser errors.

An analysis of the non-finite verbs identified by the system, but not annotated as events in TimeBank (463 cases) reveals the fact that 252 of them should have been annotated.

For the remaining 211 cases, their presence in the system's list of non-finite verbs not labelled as events in TimeBank is fully justified, as they were not supposed to be annotated according to the guidelines. As in the case of finite verbs, generic usages should not be annotated, but since no attempt is made to identify generic verbs, they are annotated in 64 cases (e.g. [6.23]). Among these, 19 instances account for verbs that form generic expressions used to elaborate in more detail on something previously mentioned (e.g. *related to* [6.24]). There are also 15 generic cases where the non-finite verbs are employed in noun phrases to qualify certain characteristics of the noun they syntactically depend on (e.g. *civil rights* ***monitoring*** *group*, ***detonating*** *cord*).

[6.23] *So for Hong Kong, it's time, as investment bankers like to* ***say****, to reposition.*

[6.24] *In addition, Hadson said it will write off about $3.5 million in costs* ***related*** *to international exploration leases where exploration efforts have been unsuccessful.*

Apart from generic usages, there are also 64 non-finite verbs occurring in article headlines, which should not receive an annotation (e.g. [6.25]). Modal and auxiliary verbs, also excluded from annotation, were identified 4 times (e.g. [6.26]). Three non-finite verbs appear in fixed phrases that do not contribute to the sentence meaning and they should not be annotated (e.g. [6.27]).

[6.25] *Qantas to* ***run*** *daily flights between Australia and India*

[6.26] *"Those fumes will exhaust themselves, and the manufacturing sector is going to start* ***getting*** *beat up in the spring."*

[6.27] *He added, "This has nothing to* ***do*** *with Marty Ackerman and it is not designed, particularly, to take the company private."*

There are 76 errors of identification produced by the syntactic parser. These

include the cases in which nouns (e.g. [6.28]), finite verbs (e.g. [6.29]), but mostly prepositions (e.g. [6.30]) are annotated as non-finite verbs.

[6.28] *And nails found in the Atlanta abortion clinic bombing are identical to those discovered at Rudolph's storage* ***shed*** *in north Carolina.*

[6.29] *Geraldine Brooks in Amman, Jordan, and Craig Forman in Cairo, Egypt,* ***contributed*** *to this article.*

[6.30] *Ranariddh's loyalists,* ***including*** *Nhek Bunchhay, his top military commander, went into hiding or fled the capital.*

It is a well known fact that annotating events is a very difficult and tedious task, even for human annotators. It is normal for annotators either to annotate extra events that should not have been annotated, or to miss out events that they probably did not consider relevant or that they simply did not notice because they were tired or bored. Therefore, it is only normal to find events that should have been annotated and were not, even if there was a human annotator and not an automatic tool performing the annotation. One should note that the percentage of finite verbs that should have been annotated (284 out of 728 analysed cases => 39.01%) is much lower than the percentage of non-finite verbs that should have been considered events (252 out of 463 analysed cases => 54.42%). This confirms the hypothesis that finite verbs capture the most important information in a sentence, and therefore the information expressed by non-finite verbs is more often not considered relevant for event annotation purposes.

6.4.3 Identification of all verbal events

The finite and non-finite verbal event identification modules described in Sections 6.4.1 and 6.4.2, respectively, are now joined together in the final verbal event identification system. When comparing the events identified by the system

against TimeBank, no distinction is made between finite and non-finite verb usages, in the sense that each verbal event identified by the system is checked against TimeBank to see if it is annotated as a verbal event in the gold standard, irrespective of the annotation indicating finite or non-finite usage.

The final system considers as verbal events all finite and non-finite verb occurrences, except any form of the verb *to be*. This identification method is evaluated against the verbal events annotated in TimeBank (i.e. those events having the **pos** attribute set on VERB). The evaluation reveals that the system performing the identification of verbal events achieves a precision of 78.51%, a recall of 96.28%, and an F-measure of 86.49%. The relatively low precision is due to over-annotation, therefore, in the future, this method will be refined in order to be able to identify generic verb usages, verbs in headlines and modals/auxiliaries not followed by a main verb, so that one can avoid their annotation. However, it should also be noted that a rather high number of verbal events missed by the human annotators are identified by the system, a fact that contributes to lowering the precision.

6.5 Annotation of verbal events

This section describes the approach taken in this work for finding the values of the attributes included in the TimeML <EVENT> tag. These attributes are:

- **eventID**: unique identification number automatically assigned to each event instance found in a text;
- **class**: each event belongs to one of the following classes: REPORTING, PERCEPTION, ASPECTUAL, I_ACTION, OCCURRENCE, I_STATE, STATE (see Section 6.2 for a detailed description of these values);

- **tense**: refers to the grammatical category of tense. This attribute can have the values: PRESENT, PAST, FUTURE, INFINITIVE, PRESPART, PASTPART, or NONE;

- **aspect**: captures the grammatical category of verbal aspect. The possible values for this attribute are: PROGRESSIVE, PERFECTIVE, PERFECTIVE_PROGRESSIVE or NONE;

- **pos**: represents the part of speech corresponding to an event. Its values can be: ADJECTIVE, NOUN, VERB, PREPOSITION, or OTHER;

- **polarity**: reveals whether the event has happened or not. The possible values for this attribute are: NEG and POS;

- **modality**: captures the modal information attached to an event (*may*, *can*, *could*, *would*, *should*, *might*).

The system is designed to identify the value of each attribute by using different information sources. The attribute **eventID** is automatically generated by the system to represent an unique identification number associated to each event. The TimeML attribute **class** receives the value associated to the verb's lemma in the annotated resource obtained as described in Section 6.3. The values of the remaining TimeML attributes are filled by using the lexico-syntactic information provided by Connexor's FDG parser. The annotation process of finite and non-finite verbal events is presented in detail below.

6.5.1 Annotation of finite verb events

At this stage each finite verb event is annotated with a TimeML <EVENT> tag, and values are assigned to the seven event attributes presented above.

The attribute **eventID** is automatically generated so that each event is uniquely identified through its ID.

The attribute **class** is the one that is most challenging to annotate among all TimeML <EVENT> attributes. This is where this work brings a novel contribution by offering the research community an annotation of all WordNet verbs with TimeML classes. This annotation can be applied to any natural language text to assign a class to each identified verbal event. To evaluate and demonstrate the usefulness of this annotation, it is applied to the finite verb events the system correctly identifies in the TimeBank articles (in terms of text span and according to the existing TimeBank annotation), i.e. 3,602 finite verb occurrences. This is done by looking up the class assigned to each finite verb identified and comparing it against the one annotated in TimeBank.

Out of the 3,602 finite verb occurrences investigated, 3,526 are found in WordNet and therefore a corresponding class exists in the annotation made as part of this research. The remaining 76 do not appear in WordNet (73 are phrasal verbs, like *succeed in*, one is an error made by the parser in identifying the lemma *placed* instead of *place*, one is an adjective wrongly annotated in TimeBank and wrongly classified by the syntactic parser as VERB - *pending*, and the last one is *nose-dive* which appears in WordNet as *nosedive*). In the case of phrasal verbs, the system automatically assigns them the class corresponding to the original verb obtained by deleting the particle, even if there is the possibility that the meaning, and consequently the attached class, may be different.

When comparing the class assigned by the system to a certain verb to the class annotated in TimeBank for that particular verb, the system correctly classifies 3,079 cases out of 3,602 (i.e. 85.48%).

The baseline which assigns to all finite verb events the most frequent class

encountered in TimeBank (i.e. OCCURRENCE) results in 1,982 correctly classified cases, and yields an accuracy of 55.02%.

To identify the upper margin of the accuracy interval, a classifier is trained by ten fold cross validation on TimeBank to assign to each verb the most frequent class assigned to it by manual annotation in TimeBank, resulting in 3,116 verb occurrences being correctly classified. This yields an accuracy of 86.50%, only 1.02% higher than the precision and recall obtained by applying the annotation developed in this work. Therefore, the conclusion is that by using the resource developed in this work one can predict the correct event class for a number of cases that is likely to be very close to the maximum number of cases that can be correctly identified by adhering to the "one class per verb" paradigm.

Section 6.4.1 mentioned that the system identified 284 finite verb occurrences that should have been annotated in TimeBank and were not. These cases were annotated manually with the corresponding event classes, and then the manual annotation was confronted with the system output for these cases, and it was revealed that in 245 cases the system assigned the correct class (i.e. 86.26%).

If instead of looking at each finite verb occurrence in TimeBank, individual verbs (lemmas) are considered, one can note that there are 769 unique finite verbs appearing in TimeBank. In 649 (i.e. 84.39%) of the cases the class assigned to a particular verb using the resource developed in this work is equal to the most frequent class assigned to it in TimeBank.

The 120 finite verbs having the class assigned by the annotation different to the most frequent class encountered in TimeBank were analysed in detail to identify what caused this disagreement.

In most cases, the verb senses used in TimeBank are different to the most frequent senses a verb is normally used with. For example, the verb *abandon*

appears twice in TimeBank (e.g. [6.31]), and both times it is annotated as ASPECTUAL. But its usage with the sense of putting an end to an event is encountered more seldom than the senses of leaving behind, of emptying, and of deserting. This verb has received the class OCCURRENCE in the annotation, but its most frequent class found in TimeBank is ASPECTUAL.

[6.31] *However, StatesWest isn't* ***abandoning*** *its pursuit of the much-larger Mesa.*

(ASPECTUAL in TimeBank)

Also, there are 31 verbs for which the most frequent class assigned in TimeBank should have been the one assigned to it in this work. This is due to errors of annotation in TimeBank. One example would be the verb *split*, which appears once in TimeBank annotated as ASPECTUAL (see [6.32]), while in the annotation it is assigned the class OCCURRENCE. Another example would be the verb *state*, which appears twice as finite verb in TimeBank and is once annotated as OCCURRENCE (see [6.33]), and once as REPORTING (see [6.34]), the most frequent class selected being OCCURRENCE. In the annotation the verb *state* is annotated as REPORTING.

[6.32] *No successor was named, and Mr. Reupke's duties will be* ***split*** *among three other senior Reuters executives, the company said.*

(ASPECTUAL in TimeBank)

[6.33] *I was pleased that Ms. Currie's lawyers* ***stated*** *unambiguously this morning... that she's not aware of any unethical conduct.*

(OCCURRENCE in TimeBank)

[6.34] *Organizers* ***state*** *the two days of music, dancing, and speeches is expected to draw some two million people.*

(REPORTING in TimeBank)

When checking all these cases of disagreement, errors have also been encountered in the annotation made as part of this work. There are 6 verbs for which the wrong class has been assigned. One example would be the verb *plan*, which was seen as describing an on-going process of devising a plan, and therefore the class OCCURRENCE was assigned to it. In TimeBank it appears 17 times denoting STATEs (e.g. [6.35]), probably being understood with the sense of having a certain intention.

[6.35] *Kuchma also **planned** to visit Russian gas giant Gazprom, most likely to discuss Ukraine's dlrs 1.2 billion debt to the company.*
(I_STATE in TimeBank)

Even if there are cases in which the annotation described in this work fails to provide the most appropriate class for a certain verb occurrence, the results obtained so far prove that this methodology for verb annotation can be useful not only in detecting the event classes for already annotated TimeBank events, but also in detecting and classifying new events missed by the TimeBank annotators.

The remaining five attributes included in the <EVENT> tag (i.e. **tense**, **aspect**, **pos**, **polarity** and **modality**) are assigned values by analysing the lexico-syntactic information provided by Connexor's FDG parser. The process relies on identifying the verb phrase a particular event is head of, and on analysing the syntactic features of this verb phrase.

The attribute **tense** is assigned the correct value in 3545 cases (accuracy of 98.41%), the grammatical **aspect** is correctly identified in 3532 cases (accuracy of 98.05%), the part of speech **pos** is obviously 100% correct due to the way

Attribute	Accuracy
class	85.48%
tense	98.41%
aspect	98.05%
pos	100%
polarity	99.11%
modality	99.61%

Table 6.1: System accuracy for annotating finite verb events

verbs are extracted from TimeBank for comparison with the system output, the **polarity** is assigned the correct value in 3570 cases (accuracy of 99.11%), and the **modality** is correctly identified in 3588 cases (accuracy of 99.61%).

Table 6.1 summarises the system accuracy for assigning values to all the attributes of the tag <EVENT> when the annotation targets only finite verb events.

6.5.2 Annotation of non-finite verb events

At this stage, the TimeML <EVENT> tag and its corresponding attributes are assigned to the non-finite verbal events identified by the system at the previous stage. The evaluation is performed only on those non-finite verbs that are also annotated in TimeBank as non-finite verbs (1,136 occurrences).

First the value of **eventID** is filled in by automatically assigning to each non-finite verb event a unique identifier.

The **class** attribute receives the value assigned to the verb in the annotation described in Section 6.3. The class assigned by the system to non-finite verb events matches the class annotated in TimeBank in 991 cases, thus the precision and recall obtained in assigning the correct class to non-finite verbal events is 87.23%. Only two verbs do not appear in WordNet (*dole* and *downsize*).

A baseline scenario could correspond to all non-finite verbal events receiving the most frequent class annotated in the corpus (i.e. OCCURRENCE), this being successful in 966 cases (i.e. 85.03%).

By applying ten fold cross validation on TimeBank (i.e. splitting all occurrences of non-finite verbal events into 10 files, then choosing for each verb its most frequent class annotated in nine files, and finally assigning the most frequent class to each verb in the remaining file), 995 instances are annotated correctly, yielding an accuracy of 87.58%. This could be seen as the upper boundary of the accuracy interval.

The system also assigns a class to those 252 instances of non-finite verbs that should have received an annotation in TimeBank (see Section 6.4.2 for more details). The result of this automatic classification process is manually evaluated, revealing 213 non-finite verb instances correctly classified (84.52%).

By examining individual verbs (lemmas) instead of verb occurrences, 470 unique non-finite verbs are found annotated as events in TimeBank. In 416 cases the class assigned to a verb in the resource presented in Section 6.3 coincides with the one most frequently annotated in the corpus, therefore there is an agreement of 88.51% between the event class associated with the verb in the annotation, and the class most frequently assigned to that verb in the TimeBank corpus.

However, in the case of 54 verbs, the most frequent class annotated in TimeBank is different to the one associated with it in the annotation. In most cases, it is just a matter of a particular sense or usage that appears more frequently in the TimeBank articles. For example, the verb *include* appears only once in TimeBank (see [6.36]), that instance being annotated as OCCURRENCE, as the verb is used in the sense of adding as part of something else or putting in as part of a set, group, or category (third sense in WordNet). Still, the verb

include is assigned the class STATE in the annotation, as it is more frequently used with the sense of having as a part or being made up out of (first sense in WordNet, see [6.37]).

[6.36] *The Internet, the global network of computers, is now far reaching into the country - extending its embrace to* ***include*** *every nook and cranny of the nation.*

[6.37] *The list* ***includes*** *the names of many famous writers.*

There are also a number of cases corresponding to errors in TimeBank, where the class should have been the one present in the annotation. One example would be the verb *quit* appearing once as a non-finite verb and wrongly annotated in TimeBank as ASPECTUAL (see [6.38]), when the class should have been OCCURRENCE.

[6.38] *If the government succeeds in seizing Mr. Antar's assets, he could be left without top-flight legal representation, because his attorneys are likely to* ***quit****, according to individuals familiar with the case.*
(ASPECTUAL in TimeBank)

In certain cases there are errors in the annotation - the class most frequently annotated in TimeBank being more suitable to characterise a verb than the one present in the annotation. One example is the verb *aim*, which is considered in the annotation a stative verb, but it is probably used more frequently as an OCCURRENCE.

The rest of the attributes included in the <EVENT> tag (i.e. **tense**, **aspect**, **pos**, **polarity** and **modality**) are assigned values by looking at the lexico-syntactic information given by the parser. The verbal group headed by a particular non-finite verb is analysed to extract values for these attributes.

Attribute	Accuracy
class	87.23%
tense	98.41%
aspect	99.38%
pos	100%
polarity	99.11%
modality	99.73%

Table 6.2: System accuracy for annotating non-finite verb events

The attribute **tense** is assigned the correct value in 1118 cases (accuracy of 98.41%), the grammatical **aspect** is correctly identified in 1129 cases (accuracy of 99.38%), the part of speech **pos** is 100% correct as only events with the part of speech VERB are extracted from TimeBank for comparison with the system output, the **polarity** is assigned the correct value in 1126 cases (accuracy of 99.11%), and the **modality** is correctly identified in 1133 cases (accuracy of 99.73%).

Table 6.2 summarises the system accuracy for finding the values of the <EVENT> attributes when the system deals only with non-finite verb events.

6.5.3 Annotation of all verbal events

This section describes the process of assigning the TimeML <EVENT> tag and its corresponding attributes to all verbal events identified according to the methodology described in Section 6.4.3.

The task of assigning values to the attributes of the tag <EVENT> associated to each verbal event is solved in the case of the attribute **class** by looking up the verb lemma in the resource developed as part of this work, while the remaining attributes are filled in by analysing the morpho-syntactic information given by the parser.

The attribute **class** is assigned a correct value in 85.57% of the cases. A baseline system that always assigns the class OCCURRENCE to each identified event would have an accuracy of 62.54%.

The attribute **tense** is assigned a correct value in 94.60% of the cases. There is a slight drop in performance (approx. 4%) when compared to the system's accuracy when dealing with finite and non-finite events individually. This fact is fully explainable by acknowledging that at this stage the distinction between finite and non-finite verbal events is completely ignored, thus allowing the acceptance of all the cases of finite verb events classified by the system or annotated in the gold standard as non-finite, and vice versa.

The system accuracy in finding values for the remaining attributes is similar to the one obtained when assigning values to the same attributes for finite and non-finite verb events separately. In the case of the attribute **aspect** the accuracy is 98.19%, for **polarity** is 99.08%, and for **modality** is 99.28%.

Table 6.3 summarises the system accuracy for assigning values to each attribute of the tag <EVENT> for all verbal events as described in this section, at the same time including the results obtained for annotating events expressed using finite and non-finite verbs individually, results which were presented in the previous sections.

6.6 Conclusions

This chapter presented efforts towards the development of a methodology to automatically identify and annotate events expressed using verbs in any natural language text.

Event Attribute	Accuracy Finite	Accuracy Non-Finite	Accuracy All Verbs
class	85.48%	87.23%	85.57%
tense	98.41%	98.41%	94.60%
aspect	98.05%	99.38%	98.19%
pos	100%	100%	100%
polarity	99.11%	99.11%	99.08%
modality	99.61%	99.73%	99.28%

Table 6.3: System accuracy for annotating all verbal events

First it addressed the process of annotation of WordNet verbs with TimeML event classes. Each WordNet verb was assigned an event class by two independent annotators who chose, according to their intuition, the TimeML event class that best covered most of that verb's important senses. The inter-annotator agreement was in terms of absolute agreement 96.80%, and in terms of kappa statistics 0.87. The cases of disagreement were clarified with a third, and, in some cases, a fourth annotation, and finally each verb was mapped to exactly one event class. The linguistic resource obtained at the end of this annotation process is very useful for assigning values to the **class** attribute of the TimeML <EVENT> tag.

An automatic method employing the resulted language resource was then developed and evaluated on TimeBank to measure its performance in identifying and annotating events expressed using verbs. The evaluation was performed separately for finite and non-finite verbs, but also for verbal events in general ignoring the finite vs. non-finite distinction.

The identification of verbal events, both finite and non-finite, relied on morpho-syntactic information provided by the syntactic parser. Having identified the extent of the verbal event, the next stage was finding the values of the attributes to be included in each verb's <EVENT> tag. Most attributes can be assigned values by analysing the morpho-syntactic information of the verb

phrase a particular verb heads. However, there is one attribute - **class** - that cannot be assigned a value using syntactical information. The semantic nature of this attribute made the process of finding its correct value very challenging for researchers. Several approaches have been developed for solving this problem, but they were either limited in the sense that they could find a value for the **class** attribute only for verbs present in TimeBank, or they were not very reliable as they could guess the correct class only in about 60% of the cases.

The approach taken in this work overcomes the disadvantages presented by previous approaches. By using the linguistic resource developed in this work, one can reliably assign an event class to any verb present in WordNet. This approach intended to be as domain independent as possible, and to cater for most of the verbs in the English language, WordNet offering almost complete coverage. In terms of unique verbs, TimeBank can provide the most frequent event class for 926 verbs, while this linguistic resource covers 11,306 verbs.

The results obtained when assigning values to the **class** attribute are above 85%, while all the other attributes can be correctly identified with an accuracy of over 94%. The result of 85% in the case of the **class** attribute is a very good result when considering that this approach assigns one class per verb. It is only normal that there are cases when the class assigned in TimeBank is different to the class present in the linguistic resource developed as part of this work. It is also normal to encounter cases where the most frequent class assigned to a verb in TimeBank does not correspond to the one associated to that verb in the resource developed here, as in certain domains only a few senses of a verb are employed, and they might not be the most frequent ones presented in linguistic dictionaries and resources.

Despite being aware that there are verbs which, given different contexts, belong to different event classes, the assumption underlying this research is that the number of such verbs is significantly lower than the number of verbs which, irrespective of their context, trigger the same event class. Granting all this, the method presented here is robust and has advantages over existing ones.

Chapter 7

Temporal Relations

7.1 Overview

This chapter addresses the identification of temporal relations that can be established among temporal expressions and events. After seeing in Chapters 4 and 5 how temporal expressions can be identified and normalised, and in Chapter 6 how events can be annotated, the next step is establishing temporal relations that hold between two events or between an event and a temporal expression.

Section 2.5 introduced the most important mechanisms that language uses to encode temporal relations: tense, aspect and time adverbials. According to the evaluation presented in Chapter 6, the system described in this thesis can identify the grammatical categories of tense and aspect with very high accuracy: 94.60% for tense and 98.05% for aspect. In addition to tense and aspect, time adverbials are another important mechanism for expressing temporal relations. Time adverbials are expressed using adverbial phrases, noun phrases, prepositional phrases and temporal clauses. The adverbial, nominal and prepositional phrases that convey the semantic role of time are considered temporal expressions, and their identification and normalisation are tasks successfully solved by the present system with an accuracy of 86.3% for identification, and 88% for normalisation

(for more details see Chapters 4 and 5). However, temporal clauses, which are an important subclass of time adverbials, are not considered temporal expressions and they have not been addressed so far in this thesis. To overcome this issue, Section 7.2 addresses the identification of temporal clauses by adopting a machine learning method that detects when ambiguous subordinators are used to introduce temporal clauses.

After possessing the capabilities to identify the most important mechanisms used by language to express temporal relations, and following a careful examination of these mechanisms, the system is augmented with modules designed to automatically identify the temporal relations that hold between any two temporal entities situated in the same sentence, between any event and the speech time (represented by the Document Creation Time in the case of news articles), as well as between two main events of two consecutive sentences.

The methodology implemented in each module is described in Sections 7.3 to 7.5. Section 7.3 presents the algorithm employed in this research to identify temporal relations between any two temporal entities located in the same sentence. Section 7.4 focuses on the methodology used for inferring the temporal relations that hold between any event and the date of the document (also known as the Document Creation Time, DCT). Section 7.5 investigates how the two main events of two successive sentences can be temporally ordered.

These modules are evaluated on TimeBank, and the results of each module are presented in the corresponding section describing its functionality in order to improve readability.

This chapter also looks at current task definitions and evaluation context concerning temporal relation identification, and proposes steps forward.

The chapter finishes with conclusions.

7.2 Identification of temporal clauses

This section describes a machine learning approach to the identification of temporal clauses by disambiguating the subordinating conjunctions used to introduce them. This method has also been described in Puşcaşu et al. (2006).

Temporal clauses are regularly marked by subordinators, many of which are ambiguous, being able to introduce clauses of different semantic roles. A corpus capturing the different usages of these subordinators has been annotated for the purpose of this work. This corpus is then used to train and evaluate personalised classifiers for each ambiguous subordinator in order to set apart temporal usages.

Temporal clauses are subordinate clauses defining the temporal context of the clause they are dependent on. As in the case of other dependent clauses, temporal clauses are regularly marked by cue phrases which indicate the relation between the dependent and main clauses. For the purpose of identifying temporal clauses, a set of cue phrases that normally introduce this type of clauses was extracted from *A Comprehensive Grammar of the English Language* (Quirk et al., 1985). In the following, it will be referred to as **the Set of Temporal Subordinators** (**STS** ={*after*, *as*, *as/so long as*, *as soon as*, *before*, *once*, *since*, *until/till*, *when*, *whenever*, and *while/whilst*}). The large majority of these cue phrases are ambiguous, being able to introduce clauses showing different semantic roles. Therefore, one cannot decide only on the basis of the cue phrase whether the clause it introduces is temporal or not. For example, a *since*-clause can either be temporal or causal. **The Set of Ambiguous Subordinators** (**SAS**) includes *as*, *as/so long as*, *since*, *when*, and *while/whilst*. This section will therefore report on an empirical investigation of all temporal connectives, as well as on the design and evaluation of statistical models associated to each ambiguous connective,

aiming to identify the cases when the introduced clauses are temporal.

The following sections will provide a grammatical overview of temporal clauses (Section 7.2.1), a description of the work involved in annotating the corpus of sentences embedding clauses introduced by ambiguous subordinators that might have temporal value (Section 7.2.2), as well as an account of the design and evaluation of the classifiers corresponding to each ambiguous subordinator (Sections 7.2.3 and 7.2.4 respectively).

7.2.1 Grammatical overview of temporal clauses

An adverbial clause of time relates the time of the situation denoted by the clause to the time of the situation expressed by the determined main clause (Quirk et al., 1985). Semantically, temporal clauses may express time position, duration or frequency. Temporal adverbial clauses generally require a subordinator. According to Quirk et al. (1985), the most common subordinators that introduce temporal adverbial clauses are: *after*, *as*, *as/so long as*, *as soon as*, *before*, *once*, *since*, *until/till*, *when*, *whenever*, and *while/whilst*.

Semantic analysis of adverbial clauses is in general complicated by the fact that many subordinators introduce clauses with different meanings, as illustrated below in the case of temporal subordinators:

- *when* used for time and concession

  [7.1] ***When*** *I awoke one morning, I found the house in an uproar.*

  (temporal *when*-clause)

  [7.2] *She paid* ***when*** *she could have entered free.*

  (concessive *when*-clause)

- *as* used for manner, reason and time

  [7.3] *The policeman stopped them* ***as*** *they were entering.*

  (temporal *as*-clause)

  [7.4] *I went to the bank,* ***as*** *I had run out of cash.*

  (reason *as*-clause)

  [7.5] *She cooks a turkey* ***as*** *her mother used to do.*

  (similarity/comparison *as*-clause)

  [7.6] ***As*** *he grew older, he was wiser.*

  (proportion *as*-clause)

- *while/whilst* used for time, concession and contrast

  [7.7] *He looked after my dog* ***while*** *I was on vacation.*

  (temporal *while*-clause)

  [7.8] ***While*** *I don't want to make a fuss, I feel I must protest at your interference.*

  (concessive *while*-clause)

  [7.9] ***While*** *five minutes ago the place had presented a scene of easy revelry, it was now as somnolent and dull as the day before payday.*

  (contrast *while*-clause)

- *since* used for reason and time

  [7.10] *I've been relaxing* ***since*** *the children went away on vacation.*

  (temporal *since*-clause)

  [7.11] *He took his coat,* ***since*** *it was raining.*

  (reason *since*-clause)

- *as long as/so long as* used for conditional and temporal clauses

  [7.12] ***As long as*** *Japan has problems with non-performing loans, the economy will not recover robustly.*

  (temporal *as/so long as*-clause)

  [7.13] *I don't mind which of them wins it* ***so long as*** *Ferrari wins.*

  (conditional *as/so long as*-clause)

The subordinators listed above together with their multiple usages are going to be disambiguated by annotating a corpus capturing their ambiguity (Section 7.2.2), and by training classifiers to distinguish between their temporal vs. non-temporal usage (Section 7.2.3). The remaining temporal subordinators are not disambiguated, as the clauses they introduce always have a temporal value, even if these clauses may also convey other meanings:

- *after*, apart from time, may indicate cause

  [7.14] ***After*** *Norma spoke, she received a standing ovation.*

- *before* may combine time with purpose, result or condition

  [7.15] *Go* ***before*** *I call the police!*

- *until/till*, apart from their main temporal meaning, may imply result

  [7.16] *She massaged her leg* ***until*** *it stopped hurting.*

- *whenever* may combine time with condition, or time with cause and condition, or time with contingency, but it is primarily used to introduce a frequency adverbial or habitual conditions

  [7.17] ***Whenever*** *I read I like to be alone.*

- *once* may imply, apart from time, contingency, condition and reason

  [7.18] *My family,* ***once*** *they saw the mood I was in, left me completely alone.*

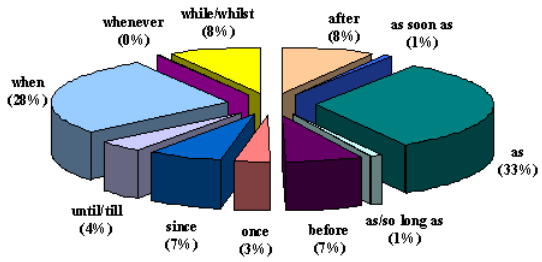

Figure 7.1: Distribution of temporal subordinators in Susanne Corpus

- *as soon as* illustrates the proximity in time of the two situations

  [7.19] ***As soon as*** *I left, I burst out laughing.*

7.2.2 Corpus annotation

This section describes the work involved in annotating a corpus of sentences embedding clauses introduced by ambiguous subordinators that might have temporal value. Each such clause is annotated as temporal or non-temporal by testing whether it answers the questions *when*, *how often* or *how long* with respect to the action of its superordinate clause.

The annotation was performed on the Susanne Corpus (Sampson, 1995), a freely available corpus developed at Oxford University consisting of 14,299 clauses explicitly annotated in terms of extent and clause type. Figure 7.1 illustrates the distribution of all temporal subordinators in the Susanne Corpus, derived by counting all the clauses introduced by each subordinator **t**$\in$**STS** (for the ambiguous subordinators no distinction was made between temporal/non-temporal usages). All **STS** subordinators account for 859 clauses in the Susanne Corpus.

The first stage of the annotation process involved extracting for each ambiguous subordinator **s**$\in$**SAS** all the sentences that included subordinate clauses initiated by **s** (either **s** was the first word in a clause, or it was preceded

only by coordinating conjunctions or modifying adverbs such as *just*, *even*, *especially*). This extraction methodology automatically excludes the cases when subordinators like *since* or *as* occupy the first position in a sentence and play the role of a preposition (example [7.20]).

[7.20] ***As*** *a detective, I always pay close attention to details.*

Out of all the levels of annotation embedded in the Susanne Corpus, only the clause and sentence boundaries were preserved. Afterwards, each clause introduced by **s** was annotated with the attribute **TEMPORAL** and assigned one of two possible values "YES" or "NO" to indicate whether the clause **s** introduces is temporal or not. The annotation was made by simply testing whether or not the subordinate clause can answer any of the questions *when*, *how often* or *how long* with respect to the action of its superordinate clause.

As there were only 9 occurrences of the subordinators *as long as* and *so long as* in the Susanne Corpus, the Reuters Corpus (Rose et al., 2002) was used to extract 50 more sentences including clauses introduced by any of the two connectives. The sentences selected from the Reuters Corpus were split into clauses and each occurrence of the connective was annotated as temporal or non-temporal. Extracting sentences from two different corpora should not pose any problems to the approach proposed here, given its general purpose nature.

The resulted corpus was then parsed using Connexor's FDG parser (Tapanainen and Jarvinen, 1997) and used for training and testing the machine learning approach described in the following section. Despite the fact that the Susanne Corpus already included manually attached part-of-speech labels, the entire corpus was parsed with an independent syntactic parser with the aim of obtaining a realistic evaluation and classifiers that can then be employed on any other type of text.

7.2.3 A machine learning approach to the identification of temporal clauses

Machine learning has been successfully employed in solving many NLP tasks, including discourse parsing. One example of employing machine learning in the disambiguation of discourse markers is presented by Hutchinson (2004). The author aims at acquiring the meaning of a wide set of discourse markers (140) and classifying them along three dimensions: polarity, veridicality and type (i.e. causal, temporal or additive). However, the temporal class of discourse markers used for training purposes included most subordinators able to introduce temporal clauses, with no attempt being made to set apart their non-temporal usages. At the same time the author excluded from his experiments discourse markers which showed a high degree of ambiguity across classes.

The machine learning method applied to the problem discussed in this section is *memory-based learning* (MBL). The MBL algorithm employed in the following experiments is the implementation of k-nearest neighbours present in the software package TiMBL (Daelemans et al., 2004).

For the purpose of identifying temporal clauses by training classifiers capable of distinguishing between temporal and non-temporal usages of ambiguous subordinators, several classes of features have been designed to characterise each training/test instance:

[I] ***Collocation features*** encode information, such as the words and their POS in a window of two words on each side of the investigated subordinator. The motivation supporting the inclusion of the surrounding words as features lies in the fact that, many times, a word's meaning can be inferred from its nearby context (Harris, 1954). The morphological information of the context words is

also useful in predicting the usage of a subordinator.

[II] ***Verb features*** The verb phrase of the subordinate clause (SubVP) and the verb phrase of the main clause (MainVP) are identified using a set of grammatical rules, and then characterised by the following features:

* MODALITY: ***future*** (*will, shall, be going to*), ***obligation/necessity*** (*must, should, have (got) to, ought to, need to, be supposed to*), ***permission/possibility/ability*** (*can, could, may, might*);
* ASPECT: ***simple, progressive, perfective, perfective progressive***;
* TENSE: ***present, past***;
* VOICE: ***active, passive***;
* POSITIVENESS: ***affirmative, negative***
* TENSE SIGNATURE: this feature conveys the representation normally used with verb phrases, that combines tense, modality and aspect (for example, it has the value ***Future Simple*** in the case of future modality and simple aspect, ***Present Progressive*** in the case of present tense and progressive aspect). It has been introduced to verify whether it produces better results than the combination of simple features characterising the verb phrases.

[III] ***Verb connection features*** This class includes:

* MainVP-SubVP: a feature that encodes the tense signatures of the two verb phrases and was included because there are many regularities manifested by the main-subordinate clause pairs corresponding to certain semantic roles (for example in the case of *when*-clauses, the correspondence ***Past Tense Simple - Past Tense Simple*** signals a temporal use)
* SAME LEMMA: a feature indicating whether the two VP lemmas are identical. The same lemma being present in both clauses may indicate contrastive -

therefore non-temporal - usage, as in example [7.21].

[7.21] *During school, Sue liked Chemistry* ***while*** *John liked Maths.*

[IV] ***Co-occurrence features*** are used to indicate whether or not, within the span covered by each feature, certain subordinator-specific phrases appear, thus pointing to a certain semantic role. The possible spans covered by these features are **the same clause** and **the main clause** span. In the case of *as*, the same clause span feature indicates whether *if* or *though* or *to whether* follow *as*, pointing to a non-temporal usage. The feature corresponding to the main clause span illustrates the presence within this span of:

* *so*, *same*, *as*, *such*, in the case of *as* (indicating non-temporal usage)
* *then*, *in that case*, for *as/so long as* (indicating non-temporal usage)
* *rather*, *however*, *therefore*, *how*, in the case of *since* (indicating non-temporal usage)
* *then*, *always*, *never*, *often*, *usually*, *every*, in the case of *when* (indicating temporal usage)
* *yet*, *besides*, *on the other hand*, *instead*, *nevertheless*, *moreover*, in the case of *while/whilst* (indicating non-temporal usage)

[V] ***Structural feature*** denotes the position of the subordinate clause with respect to the matrix clause (before, after or embedded), also indicating the presence/absence of punctuation signs between the two clauses.

[VI] ***FDG-relation*** contains information provided by the Connexor FDG parser that predicts the type of relation holding between the subordinate and matrix clauses. This information is normally attached by the parser to the verb phrase of the subordinate clause.

The classes of features described so far were defined so that their values can be automatically extracted from any text analysed with Connexor's FDG Parser. These features were employed in the experiments described in the following section.

7.2.4 Experiments

To identify the most appropriate model for the disambiguation of each subordinator, several feature combinations have been evaluated using the machine learning method described in the previous section. Each model was evaluated with the *leave-one-out* approach, similar to 10-fold cross-validation, a reliable way of testing the performance of a classifier. The underlying idea of the *leave-one-out* approach is that every instance in turn is selected once as a test item, and the classifier is trained on all remaining instances.

For each connective the baseline was considered to be a classifier that assigns to all instances the class most commonly observed among the annotated examples. Twelve different models have been evaluated to compare the relevance of various feature classes to the classification of each temporal connective. The evaluated models are described in detail in the following:

* **MainVP (Tense Signature only)** This model is trained using only the tense signature of the main clause's verb phrase.

* **MainVP (All features)** The five characteristics included in the verb feature class (modality, aspect, tense, voice, positiveness) of the main clause VP are used.

* **SubVP (Tense Signature only)** The model is trained using only the tense signature of the subordinate clause's VP.

* **SubVP (All features)** The five simple features of the VP corresponding to the subordinate clause are used for training.

* **BothVP (MainVP + SubVP)** All features characterising the two verb phrases are included in this model.

* **BestVP** This model designates the best performing VP model observed so far.

* **VPCombi (BestVP + VPConnection)** The best performing verb phrase model, together with the verb connection features are employed at this stage.

* **VPCombi + Collocation features** This model comprises the combination of VP features, as well as the features characterising the context of the connective.

* **VPCombi + Co-occurrence features** This model is trained with the VPCombi model features combined with the co-occurrence features of the corresponding connective.

* **VPCombi + Structural feature** The VPCombi model together with the structural feature form the present model.

* **VPCombi + FDG-relation** This model comprises the VPCombi model features and the FDG-relation feature capturing the functional dependency holding between the two clauses.

* **VPCombi + Best combination** The present model embeds the features of the VPCombi model, as well as the best combination of features chosen from the four feature classes: collocation, co-occurrence, structural and FDG-relation.

* **All** This model is trained with all feature classes described in Section 7.2.3.

Table 7.1 captures the accuracy of all the models presented above for the task of classifying each connective use as temporal or not. Figures in bold indicate

the best performing model per connective.

CONNECTIVE / CLASSIFIER	AS	AS LONG AS SO LONG AS	SINCE	WHEN	WHILE WHILST
Baseline	67.38%	73.21%	85.00%	86.86%	52.77%
MainVP (Tense Signature only)	74.19%	64.28%	96.66%	84.74%	58.33%
MainVP (All features)	76.70%	64.28%	96.66%	84.32%	47.22%
SubVP (Tense Signature only)	70.25%	78.57%	90.00%	90.67%	75.00%
SubVP (All features)	74.55%	80.35%	96.66%	87.28%	75.00%
BothVP = MainVP + SubVP	81.72%	75.00%	95.00%	91.94%	72.22%
BestVP = MAX(MainVP, SubVP, BothVP)	81.72%	80.35%	96.66%	91.94%	75.00%
VPCombi = BestVP + VPConnection	81.72%	**82.14%**	95.00%	92.37%	76.38%
VPCombi + Collocation features	86.02%	67.85%	95.00%	89.40%	65.27%
VPCombi + Co-occurrence features	81.72%	82.14%	96.66%	**92.79%**	81.94%
VPCombi + Structural feature	81.00%	69.64%	96.66%	90.25%	83.33%
VPCombi + FDG-relation	83.87%	76.78%	95.00%	90.67%	79.16%
VPCombi + Best combination	**88.17%**	82.14%	**98.33%**	92.79%	**84.72%**
All features	86.37%	71.42%	98.33%	91.10%	73.61%

Table 7.1: Accuracy of various classifiers in discovering temporal usages of ambiguous connectives

The best model for *as* includes the grammatical features of the two verb phrases, the verb phrase connection features, the collocation and functional dependency features, achieving an accuracy of 88.17% in distinguishing between temporal and non-temporal usages of *as*. The collocation features proved to be useful only in the case of *as*, due to many cases where the connective was preceded by another *as* followed by an adjective or an adverb, signalling non-temporal usage.

In the case of *as/so long as*, the best model with an accuracy of 82.14% comprises the features characterising the subordinate clause VP and the VPConnection. In addition, the same performance is obtained by two other classifiers (VPCombi + Co-occurrence features and VPCombi + Best combination), but they are more complex, and therefore require more computational power, so the simplest classifier is preferred.

Since is best dealt with by the VP features of the main clause, combined with VPConnection, structural and co-occurrence features, and the correct distinction

between temporal and non-temporal *since* is made in 98.33% of the cases. The verb phrase of the main clause proves to be very important in the classification of *since*, because a temporal *since*-clause generally requires the Present or Past Perfective in the matrix clause.

The best classifier for *when* combines the features corresponding to both verb phrases, VPConnection and co-occurrence with an accuracy of 92.79%. The same performance is obtained by a more complex classifier (VPCombi + Best combination), but again preference is given to the simplest model.

In the case of *while/whilst*, the best performing model includes the subordinate clause's VP, the VPConnection, the structural and the FDG-relation features, and its accuracy is 84.72%.

An examination of the errors revealed two main causes. On the one hand, there are cases when the syntactic parser fails in identifying verbs, thus leading to erroneous values being attached to the features attached to the verb phrases of the two clauses. On the other hand, due to the fact that the classifiers do not rely on a semantic analysis of the clauses connected by a certain connective, two syntactically similar pairs of main-subordinate clauses can lead to the same class being assigned to the connective lying between them. This lack of semantic information leads to many classification errors, as instanced below:

[7.22] ***As*** *she held her speech, he thought about what they had spoken before.*

(temporal *as*-clause, correctly classified as temporal)

[7.23] ***As*** *we expected, my uncle recovered fast.*

(non-temporal *as*-clause, but incorrectly classified as temporal)

The experiments presented in this section demonstrate a variation in performance between different subordinators, with the classifiers for *as* and *while/whilst* at

21%, respectively 32%, above the baseline. The macro average accuracy across all investigated connectives is 89.23%, significantly above the average baseline of 73.04%. It is possible that an increased size of the training set could lead to an improved performance. In the case of all connectives, the most informative features have proved to be those derived from the verb phrases of the main and subordinate clauses.

Temporal clauses are used to establish temporal relations between events, but also to bring into focus a novel temporal referent whose unique identifiability in the reader's memory is presupposed, thus updating the current reference time (Reichenbach, 1947). The ability to identify temporal clauses will be exploited in the development of the modules presented in the following two sections, modules that deal with the identification of intra-sentential temporal relations and of temporal relations holding between events and the DCT.

7.3 Identification of intra-sentential temporal relations

A detailed investigation of all the temporal relations annotated in TimeBank 1.2 (6418 TLINKs) reveals that approximately 60% of these relations link two temporal entities (events or TEs) situated in the same sentence. Another approximately 20% of the TLINKs annotated in TimeBank are relations between temporal entities and the Document Creation Time (DCT). The remaining percentage of temporal relations (approximately 20%) hold between temporal entities situated in different sentences. Table 7.2 captures the different types of temporal relations classified according to the categories and the relative location in the text of the two connected temporal entities.

Type of TLINK	Number of TLINKs
TLINKs between two events situated in the same sentence	2368
TLINKs between an event and a TE from the same sentence	1339
TLINKs between two TEs situated in the same sentence	28
TLINKs between an event and the DCT	1275
TLINKs between a TE and the DCT	71
TLINKs between events situated in consecutive sentences	573
TLINKs between events situated in different sentences (more than one sentence apart)	540
TLINKs between an event and a TE located in different sentences	183
TLINKs between two TE situated in different sentences	41

Table 7.2: Distribution of temporal relations in TimeBank 1.2

The fact that human annotators link most frequently temporal entities situated in the same sentence via temporal relations is perfectly understandable, as local context provides many explicit clues as to what temporal relation holds between two entities. By broadening the context and increasing the textual distance between two entities, not only does one require more inferences to decide upon the temporal relation, but at the same time the chances of the two entities being temporally unrelated increase.

The methodology adopted in this research for the identification of intra-sentential temporal relations closely follows human behaviour when deciding the temporal relation holding between two entities, and exploits explicit textual evidence encoded mainly in the syntax of a given sentence. The approach taken in the present work is thus knowledge-based and relies on a complex syntactic analysis of the text. It employs sentence-level syntactic trees and a bottom-up propagation of the temporal relations between syntactic constituents, by analysing syntactic and lexical properties of the constituents and of the relations between them. A temporal inference mechanism is afterwards employed to relate

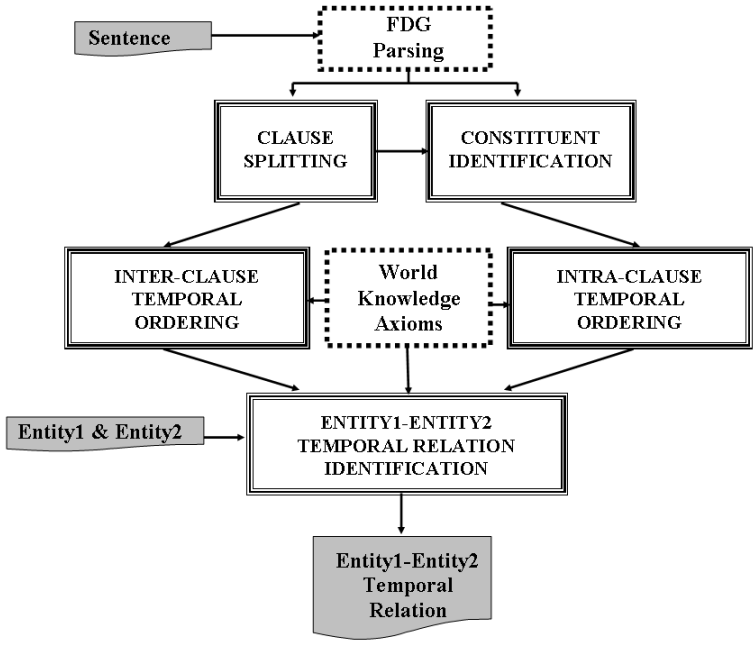

Figure 7.2: Processing stages for the intra-sentential temporal relation identifier

any two targeted temporal entities to their closest ancestor and then to each other. Conflict resolution heuristics are also applied whenever conflicts occur. Using this approach, one can discover temporal relations between any two temporal entities, events or TEs, whenever the two entities are situated in the same sentence.

Figure 7.2 depicts the processing stages involved in the identification of the temporal relation given the two temporal entities and the sentence they are in.

The sentence is first annotated with morpho-syntactic and functional dependency information by employing Connexor's FDG parser (Tapanainen and Jarvinen, 1997). For newspaper articles this parser reports a success rate of 96.4% at morpho-syntactic level and an f-measure of 91.45% when attaching heads in a dependency relation.

A clause splitter previously developed by the author (Puşcaşu, 2004a) is then used to detect clause boundaries and to establish the dependencies between the resulted clauses by relying on formal indicators of coordination and subordination and, in their absence, on the functional dependency relation predicted by the FDG parser. This clause splitter was evaluated on the Susanne Corpus (Sampson, 1995) and the F-measure for the identification of complete clauses was 81.39%.

Each clause is then individually processed to obtain a temporal ordering of the clause constituents (**intra-clausal temporal ordering**), and afterwards a similar temporal ordering process is applied to each pair of clauses involved in a dependency relation (**inter-clausal temporal ordering**). At the end of this process, each branch of the syntactic tree connecting a non-root node with its antecedent is labelled with a temporal relation. An example of a labelled syntactic tree corresponding to the sentence [7.24] can be found in Figure 7.3.

[7.24] *An IBM spokeswoman said the company told customers Monday about the bugs and temporarily stopped shipping the product.*

The final stage involves the detection of the temporal relation between two temporal entities, both situated in the sentence processed as above. The following sections describe each of the three stages involved in finding the intra-sentential temporal relations between any two temporal entities.

7.3.1 Intra-clausal temporal ordering

This stage begins by identifying the set of temporally relevant constituents present in each clause by examining the morpho-syntactic information provided by Connexor's FDG parser. The temporally relevant clause constituents are considered to be: the verb phrase VP, the noun phrases NPs, the prepositional phrases PPs, the non-finite verbs and the adverbial temporal expressions present

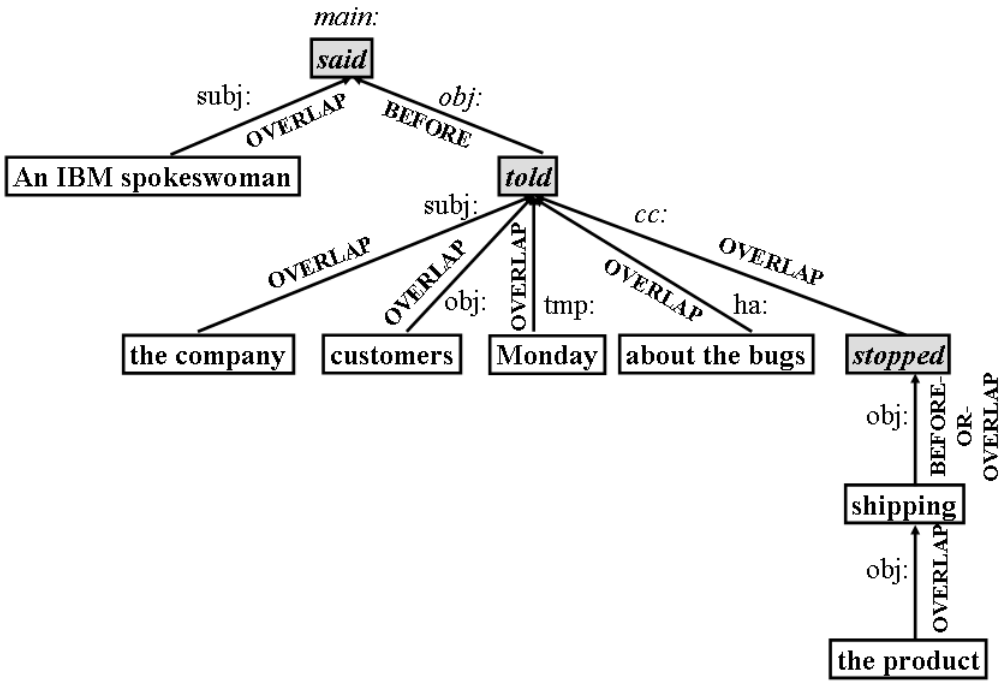

Figure 7.3: Syntactic tree labelled with temporal relations

in the analysed clause.

The identified constituents and the syntactic tree of the corresponding clause are afterwards employed in a recursive bottom-up process of finding the temporal order between directly linked constituents. The leaf nodes are first linked to their syntactic antecedents [1], then by going up the syntactic tree each non-leaf and non-root node is linked to its antecedent until there is a path of temporal relations from each leaf node up to the root of each clause's syntactic tree - the central verb phrase. Each constituent is linked only with the constituent it syntactically depends on using one of the predefined temporal relations.

The temporal relation between two constituents is decided on the basis of generally applicable heuristics that involve parameters such as: the semantic properties of the two constituents' heads (whether their root forms denote

1. The syntactic antecedent of a leaf node is the node that immediately dominates the leaf node in the syntactic tree.

reporting or aspectual start/end events - this is decided by consulting lists of reporting/aspectual start/aspectual end events extracted from the annotation described in Chapter 6), the types of the two constituents, the syntactic relation holding between them, the presence of certain temporal signals (e.g. prepositions like *before*, *after*, *until*, *since*), the tense of the clause's verb phrase, and the temporal relation between any of the clause's temporal expressions and the DCT. The default temporal relation holding between any constituent and its syntactic antecedent is OVERLAP, but this relation is changed whenever any of the parameters enumerated above indicate a different relation. For example, given the clause *he likes the silence before the storm*, the relation between *the storm* and *the silence* is imposed by the temporal preposition *before*, *the storm* being thus temporally located AFTER *the silence*.

The rules involved in linking two constituents situated in the same clause will be illustrated on the case of two events EVENT_1 and EVENT_2, where EVENT_2 is the direct object of EVENT_1. The system implements the following rules:

- if EVENT_1 is a REPORTING event, then EVENT_2 BEFORE EVENT_1;
- if EVENT_1 is an ASPECTUAL event, then EVENT_2 OVERLAP EVENT_1;
- if EVENT_1 is a PERCEPTION event, then EVENT_2 OVERLAP EVENT_1;
- if EVENT_1 is an OCCURRENCE event and EVENT_2 is an infinitive verb, then EVENT_2 AFTER EVENT_1;
- if EVENT_1 is an OCCURRENCE event and EVENT_2 is a progressive verb, then EVENT_2 OVERLAP EVENT_1;
- if EVENT_1 is an OCCURRENCE event and EVENT_2 is a noun and EVENT_1

is a synonym of the verb *to cause*[2], then EVENT_2 AFTER EVENT_1;

- if EVENT_1 is an OCCURRENCE event and EVENT_2 is a noun and EVENT_1 is not a cause event, then EVENT_2 OVERLAP EVENT_1;
- if EVENT_1 is a STATE event, then EVENT_2 OVERLAP EVENT_1, irrespective of EVENT_2 being a noun or an infinitive or progressive verb.

Following the above recursive process of linking any two syntactically related clause constituents via a temporal relation, there is a path of temporal relations from any clause constituent to the clause's central VP. After this process has been applied to each clause in a given sentence, the next stage is inter-clausal temporal ordering.

7.3.2 Inter-clausal temporal ordering

At this stage, each pair of clauses involved in a dependency relation are temporally ordered. The information provided by the tenses of their VPs and by the dependency relation holding between the two clauses is very important for this process. The underlying hypothesis is that the clause binding elements and the tenses of the two central VPs provide a natural way to establish temporal relations between two syntactically related clauses.

The property of the superordinate clause's verb of being a reporting, aspectual or perception event is also relevant at this stage. The object clause of a reporting event is typically situated prior to the reporting event on a timeline except the cases where the object clause talks about a future event either via tense or by

2. Causality is dealt with in this work using a simplistic approach, by considering cause events to be expressed by the verb *to cause* or by any of its synonyms present in the *Roget's 21st Century Thesaurus*.

mentioning TEs situated in the future with respect to the DCT. Aspectual events refer to different stages in the evolution of an event, thus overlapping temporally with the event they take as object. Perceptual events also overlap the event they take as object, as the perceived event happens roughly at the time when it is perceived.

The temporal expressions modifying the verb phrases of the two clauses involved in a syntactic relation can also help in relating the two clauses temporally.

Given for example the case of two clauses CLAUSE_1 and CLAUSE_2, where CLAUSE_2 is a temporal clause subordinated to CLAUSE_1 (for more details on how temporal clauses are identified in this work, see Section 7.2). This work focuses on temporal clauses introduced by one of the subordinators included in the Set of Temporal Subordinators **STS** introduced in Section 7.2 (**STS** ={*after*, *as*, *as/so long as*, *as soon as*, *before*, *once*, *since*, *until/till*, *when*, *whenever*, and *while/whilst*}). The system implements the following rules in inferring the temporal relation between a temporal clause (CLAUSE_2) introduced by one of the above subordinators and its superordinate clause (CLAUSE_1):

- **Rule 1:** the temporal relation between a temporal clause introduced by *after* and its main clause is BEFORE (CLAUSE_2 BEFORE CLAUSE_1);

- **Rule 2:** according to Thompson (2005), temporal clauses introduced by *as* force the adjunct event time to be interpreted as simultaneous with the time of the matrix event (the event of the superordinate clause CLAUSE_1), therefore the temporal relation between a temporal *as*-clause and its superordinate clause is OVERLAP (CLAUSE_2 OVERLAP CLAUSE_1);

- **Rule 3:** the temporal relation between a clause introduced by *as long as* or *so long as* and its main clause is OVERLAP (CLAUSE_2 OVERLAP CLAUSE_1);
- **Rule 4:** in the case of *as soon as*, the action in the subordinate clause is temporally located BEFORE the action described by the main clause (CLAUSE_2 BEFORE CLAUSE_1);
- **Rule 5:** a temporal clause introduced by *before* is always temporally AFTER its matrix clause (CLAUSE_2 AFTER CLAUSE_1);
- **Rule 6:** *since* temporal clauses are temporally BEFORE their main clauses (CLAUSE_2 BEFORE CLAUSE_1);
- **Rule 7:** the temporal relation between a temporal clause introduced by *until* or *till* and its superordinate clause is AFTER (CLAUSE_2 AFTER CLAUSE_1);
- **Rule 8:** if the temporal clause CLAUSE_2 is introduced by *when*, the following parameters are important for deciding the temporal relation between the two clauses: the tense and aspect of the two verb phrases, the aspectual event types of the main events heading each clause, as well as the relative textual position of the subordinate clause with respect to the main clause.
  - **Rule 8.1:** If the aspect of the main verb phrase is Perfect, the presence of the aspectual morpheme *have* orders the Event Time of the main clause event as preceding the Reference Time that is normally modified by the temporal clause, thus situating the main event time before the subordinate event time. In this case the temporal relation between CLAUSE_2 and CLAUSE_1 is AFTER (CLAUSE_2 AFTER CLAUSE_1).
  - **Rule 8.2:** In the absence of the aspectual morpheme *have* from both clauses (i.e. both verb phrases are characterised by simple or

progressive tenses), *when*-clauses are ambiguous in that they permit either a simultaneous or non-simultaneous reading. If the grammatical aspect of either verb phrase is Progressive, or the aspectual class of either head event is STATE, then the temporal relation between CLAUSE_2 and CLAUSE_1 is OVERLAP (CLAUSE_2 OVERLAP CLAUSE_1).

- **Rule 8.3:** Otherwise, in the absence of both the Perfective and the Progressive aspect from the two clauses, a *when* clause preceding the main clause has only a non-simultaneous reading, according to Thompson (2005). In such cases the temporal relation between CLAUSE_2 (the *when*-clause) and CLAUSE_1 is BEFORE (CLAUSE_2 BEFORE CLAUSE_1).
- **Rule 8.4:** For any remaining cases (simple tenses, non-stative events, and the *when*-clause is either embedded or after the main clause), it will be assumed that the temporal relation is also BEFORE, even if in reality this is not always the case, but due to the limitations of the system in accessing deeper semantic information, this will be the default behaviour (CLAUSE_2 BEFORE CLAUSE_1).

- **Rule 9:** temporal clauses introduced by *whenever* receive the same treatment as *when*-clauses;

- **Rule 10:** temporal *while*-clauses are contemporaneous with their matrix clauses, the temporal relation that applies to them being OVERLAP (CLAUSE_2 OVERLAP CLAUSE_1).

At the end of the intra-clausal and inter-clausal processing stages, each branch of the syntactic tree connecting a non-root node with its antecedent is labelled with a temporal relation, like in the example present in Figure 7.3. The next

stage described in the following section involves using this labelled syntactic tree to infer the temporal relation between any two temporal entities belonging to the same sentence.

7.3.3 Identification of the temporal relation holding between two co-sentential temporal entities

This stage involves retrieving the temporal relation between any two temporal entities situated in the sentence processed as above. The two entities are first tested to determine if they comply with world knowledge axioms that would predict their temporal relation. For example, if one entity is a TE that refers to a date that is previous to the DCT, and the other entity is an event expressed via a future tensed verb, then the temporal relation between the event and the TE is obviously AFTER. If no axiom applies to the two entities, a temporal reasoning mechanism is employed to relate the two targeted temporal entities to their closest syntactic ancestor, and then to each other.

If conflicts occur in relating one entity to the ancestor, priority is given to the relation linked to the entity, but if the conflict is between the temporal relations of the two entities with the ancestor, the relation of the entity situated higher in the functional dependency tree with the ancestor wins.

7.3.4 Evaluation

The system implementing the methodology described above for the identification of intra-sentential temporal relations took part in the TempEval evaluation exercise, being evaluated along with other systems for three different tasks, as described in Puşcaşu (2007b). The first task addressed the temporal relations

	BEFORE	OVERLAP	AFTER	BEFORE-OR-OVERLAP	OVERLAP-OR-AFTER	VAGUE
BEFORE	1	0	0	0.5	0	0.33
OVERLAP	0	1	0	0.5	0.5	0.33
AFTER	0	0	1	0	0.5	0.33
BEFORE-OR-OVERLAP	0.5	0.5	0	1	0.5	0.67
OVERLAP-OR-AFTER	0	0.5	0.5	0.5	1	0.67
VAGUE	0.33	0.33	0.33	0.67	0.67	1

Table 7.3: Relaxed scoring scheme for partial matches

holding between time and event expressions situated in the same sentence. Only the events that occurred twenty times or more in TimeBank were considered (this set of events is referred to as the Event Target List or ETL).

The TempEval training and test data consists of all the news articles included in the TimeBank corpus, only that for TempEval they were annotated with a simplified version of TimeML. The TimeML TIMEX3 and EVENT tags apply to the same TEs and events annotated in TimeBank, with a minor modification in the case of the EVENT tag that now merges the information originally encoded in TimeBank in both the EVENT and the MAKEINSTANCE tags. There is also an extra attribute added to the EVENT tag - **mainevent** - indicating whether or not an event is the main event of a sentence. The rest of the attributes and the values associated to the TIMEX3 and EVENT tags are the same as in the TimeBank annotation. The TempEval TLINK tag is a simplified version of the TimeML TLINK tag. Compared to the original set of 14 temporal relations defined by TimeML, TempEval uses only the following five: OVERLAP, BEFORE, AFTER, BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER. There is also a VAGUE relation used in the TempEval annotation for those cases where no particular relation can be established.

The TempEval data is split into a set of 163 articles for training and 20

Intra-sentence temporal ordering	STRICT SCORE			RELAXED SCORE		
	P	R	F	P	R	F
BASELINE	0.49	0.49	**0.49**	0.51	0.51	**0.51**
TempEval-TRAIN	0.65	0.65	**0.65**	0.68	0.68	**0.68**
TempEval-TEST	0.62	0.62	**0.62**	0.64	0.64	**0.64**
TimeBank	0.65	0.65	**0.65**	0.67	0.67	**0.67**

Table 7.4: System results for intra-sentential temporal ordering of Event-TE pairs

articles for testing. These are the 183 articles that constitute TimeBank 1.2. All articles include the following information: sentence boundaries, temporal expressions (with a special label indicating the DCT), events (only those whose root form occurs in the ETL are annotated), and the temporal relations between each annotated event and the time expressions located in the same sentence, between each annotated event and the DCT, and also between the main verbs of any two consecutive sentences. In the case of the test data, the TLINK tags indicate the two entities involved in the temporal relation, but leave the value of the temporal relation unspecified.

TempEval uses as evaluation metrics precision, recall and f-measure, as well as two scoring schemes: strict and relaxed. The strict scoring scheme counts only exact matches, while the relaxed one gives credit to partial semantic matches too, according to the values presented in Table 7.3.

Table 7.4 presents the detailed evaluation results of the present system corresponding to the baseline, TempEval training data, TempEval test data and the entire TimeBank corpus. The baseline is established by the most frequent temporal relation encountered in the training data for the targeted relation type. In the case of event-TE intra-sentential temporal relations, this relation is OVERLAP.

According to the TempEval evaluation results (Verhagen et al., 2007), the

TEAM	STRICT SCORE			RELAXED SCORE		
	P	R	F	P	R	F
CU-TMP	0.61	0.61	0.61	0.63	0.63	0.63
LCC-TE	0.59	0.57	0.58	0.61	0.60	0.60
NAIST	0.61	0.61	0.61	0.63	0.63	0.63
USFD	0.59	0.59	0.59	0.60	0.60	0.60
WVALI	**0.62**	**0.62**	**0.62**	**0.64**	**0.64**	**0.64**
XRCE-T	0.53	0.25	0.34	0.63	0.30	0.41

Table 7.5: Official TempEval results for intra-sentential temporal ordering of Event-TE pairs

system developed as part of this thesis achieved the highest strict and relaxed scores for the task of intra-sentential temporal ordering. The official results of all participating systems can be found in Table 7.5. The participating systems are: CU-TMP (University of Colorado at Boulder), LCC-TE (Language Computer Corporation), NAIST (Nara Institute of Science and Technology), USFD (University of Sheffield), WVALI (the system presented in this thesis), XRCE-T (XEROX Research Centre).

The identification of temporal relations is not a straightforward task, its difficulty being also proven by the relatively low inter-annotator agreement achieved for the manual annotation of temporal relations. Given the initial TimeML set of 14 temporal relations, the kappa statistics measured in the annotation of TimeBank with temporal relation types was 0.71 [3]. Despite the fact that the TempEval efforts were directed towards simplifying the task by defining a reduced set of temporal relations (OVERLAP, BEFORE, AFTER, BEFORE-OR-OVERLAP and OVERLAP-OR-AFTER), it was surprising to see lower inter-annotator agreement figures. The inter-annotator agreement for the task of intra-sentential temporal relation annotation was in terms of kappa agreement

3. http://timeml.org/site/timebank/documentation-1.2.html

0.54, while in terms of percentage of cases where annotators agree (precision) it was 69%.

During the annotation of data for TempEval, the task organisers noticed a small number of cases tagged by humans using the disjunctive relation labels BEFORE-OR-OVERLAP and OVERLAP-OR-AFTER. This was surprising especially as these labels were added to facilitate annotation in the cases when annotators faced difficulties in deciding between two temporal relations. The TempEval organisers also noticed far more disagreement than agreement in the case of the disjunctive relation types, thus raising the question of whether these labels are truly useful in a temporal relation annotation scheme. The poor distribution of these disjunctive labels in the training data, as well as the observed low system performance on these labels due to unclear guidelines as to when these labels should be used, all suggest using only three labels (OVERLAP, BEFORE and AFTER) in the task of temporal relation identification. Several other authors (Verhagen et al., 2009; Lee and Katz, 2009) indicate that such a simplification would help drive research forward in the area of temporal relation identification.

Another important problem identified during TempEval involved the definition of the tasks. The low inter-annotator agreement observed during data annotation not only showed that humans cannot agree on the temporal relation to be assigned to a pair of temporal entities, but it was also an indicator of the performance level that can be expected from an automatic system that tries to solve the tasks at hand. It was proved once again that it is very complex to ask humans, let alone machines, to annotate temporal relations without imposing any constraints or predefined structure to the tasks, or without creating detailed guidelines. The tasks of identifying temporal relations, in the manner that they have been defined so far, give too much freedom and too little guidance

to the annotators. Therefore, another lesson learned from TempEval is that task decomposition is extremely advisable. Not only will clearer and focused task definitions facilitate a more reliable data annotation process, but it will also allow better system evaluation and error analysis in order to identify task-specific problems and solutions.

To overcome these problems, this thesis proposes the following simplifications in the case of intra-sentential temporal relations:

1. Annotate temporal relations using only the core set of labels: OVERLAP, BEFORE and AFTER.

2. Decompose the intra-sentential temporal relation identification problem into smaller subtasks, including:

- **Identification of intra-clausal temporal relations between a TE and a governing nominal event**

  This subtask would target temporal relations holding between a temporal expression and a nominal event, given that the nominal event syntactically dominates the temporal expression.

- **Identification of intra-clausal temporal relations between a TE and a dominating verbal event**

  This subtask would look at temporal relations holding between a temporal expression and a verbal event, given that the verbal event governs the temporal expression.

- **Identification of intra-clausal temporal relations between two events involved in a syntactic dependency relation**

  This subtask would investigate temporal relations that hold between two events that are involved in a syntactic dependency relation, given that the two events are located in the same clause (the analysis should be guided

by the syntactic relation between the two events, but also by the POS and class of the two events).

- **Identification of inter-clausal temporal relations between the central events of two clauses involved in a syntactic dependency relation**

  This subtask would target the temporal relations holding between the central events of two clauses involved in a syntactic dependency relation (for each syntactic relation holding between two clauses, this task would involve a detailed analysis of the parameters relevant to the identification of the temporal relation between them).

In the remainder of this section the system is evaluated on each of the above mentioned subtasks.

Identification of intra-clausal temporal relations between a TE and a governing nominal event

For this task, the accuracy of the system is measured using two different settings. The data used for evaluation is a simplified version of the TempEval data in the sense that each relation initially annotated with BEFORE-OR-OVERLAP or OVERLAP-OR-AFTER is now converted by a human annotator into one of the three core relations: OVERLAP, BEFORE or AFTER.

In the first evaluation setting, only the temporal expressions directly dependent on a noun event are looked at. Direct dependency includes the cases when the temporal expression modifies the noun either directly (*the Monday lecture*) or via a preposition (*the lecture on Monday*). The system identifies the correct temporal relation between the TE and the nominal event it directly

modifies in 100% of the cases (45 out of 45 cases are correctly identified). This high accuracy is not surprising given the fact that in the absence of a preposition the TE that modifies the noun indicates the time when the noun event took place, thus always yielding the OVERLAP relation between the two temporal entities. Whenever a preposition intervenes between the TE and the noun it modifies, this preposition indicates the temporal order of the two entities.

The second evaluation setting allows any number of dependency links on the syntactic path between the TE and the noun it directly or indirectly modifies. This means that the TE is syntactically governed by the nominal event, any number of words (including 0) being allowed on the syntactic path linking the two entities. The entities are only restricted to being situated in the same clause. The TempEval data includes 73 such cases, and the system identifies the correct temporal relation for 68 of them with an accuracy of 93.15%. Errors are caused by the system's lack of semantic knowledge and by the syntactic parser in building the dependency tree.

Identification of intra-clausal temporal relations between a TE and a dominating verbal event

Similar settings as in the case of nominal events are used here. The first setting evaluates only the assignment of temporal relations for TEs and verbal events in cases where the TE is directly linked to the verbal event. There are 330 such cases annotated in the TempEval data, the evaluation showing that the system is able to correctly identify the temporal relation holding between 304 verb - TE pairs. Therefore, the system performance in this setting is 92.12%. The largest source of errors (46.15%) arises from wrong PP-attachment in the cases where

the TE is preceded by a preposition and the resulted PP is incorrectly linked to that verbal event. Another important source of disagreement is caused by wrong human annotations (26.92%). Vagueness and inaccessible semantic information account for the remaining errors.

The second setting relaxes the constraints imposed on the dependency between the TE and the event, allowing syntactic paths of variable lengths between the two temporal entities. Out of 520 cases, the system correctly assigns a temporal relation to 441, the accuracy being 84.80%. The errors produced by the system are mainly due to the lack of semantic information and world knowledge involving the words situated on the path between the TE and the verbal event. Some errors are introduced by the syntactic parser due to generating incorrect syntactic trees.

Identification of intra-clausal temporal relations between two events involved in a syntactic dependency relation

The data used for this task is generated from the original TimeML annotation of TimeBank 1.2. The original set of 14 temporal relations is narrowed down to the core set of only three temporal relations (OVERLAP, BEFORE, AFTER). This is achieved by automatically mapping:

- SIMULTANEOUS, INCLUDES, IS_INCLUDED, DURING, DURING_INV, BEGINS, BEGUN_BY, ENDS, ENDED_BY and IDENTITY to OVERLAP;
- BEFORE and IBEFORE to BEFORE;
- AFTER and IAFTER to AFTER.

According to the syntactic parser employed in this work, there are 1615

event pairs located in the same clause and involved in a syntactic relation. The annotation present in TimeBank shows that in 1206 of the cases there is no temporal relation annotated for the syntactically related event pairs. There are only 409 event pairs that are linked through a temporal relation in TimeBank. Provided that this work does not focus on investigating which entities should be linked via a temporal relation, but its main aim is to identify the temporal relations between given pairs of temporal entities, only the pairs of syntactically related events involved in temporal relations according to the TimeBank annotation are further considered. A closer look at these pairs reveals that the most frequent syntactic relation linking these pairs of events is the OBJ relation indicating the fact that one event is the direct object of the other. The OBJ relation is present in 55.50% of the cases (227 pairs out of the 409 annotated in TimeBank).

Given its prominence among intra-clausal event-event syntactic relations, the OBJ relation will represent the focus of the following evaluation. The system that implements the methodology detailed in Section 7.3.1 correctly identifies the temporal relation between an event and its direct object subordinate event in 81.49% of the cases (185 out of 227 cases). The system encounters problems in the case of noun events being the direct object of OCCURRENCE events. These problems arise from the system's lack of semantic and world knowledge. In example [7.25], the event *calls* is the direct object of the event *return* and is temporally situated BEFORE it on a timeline. The system erroneously labels the temporal relation between the two events as OVERLAP.

[7.25] *Crane officials didn't <return> phone <calls> seeking comment.*

Another problem that appears in the case of two events linked by the OBJ dependency relation applies to aspectual events and their direct object

dependents. The human annotation in such cases is not consistent. For example in the case of the pair <*stop*> <*originating*> the temporal relation present in TimeBank is OVERLAP. However, in the similar case of the pair <*stopped*> <*providing*>, the annotated temporal relation between *providing* and *stopped* is BEFORE. Since the TimeML guidelines are rather unclear and lack a high level of detail, it is not surprising that many inconsistencies can be found in the annotation. This is one good reason to split the temporal relation annotation task into smaller subtasks, and create detailed annotation guidelines for each subtask to achieve a high level of inter-annotator agreement, thus allowing specialised automatic modules to be efficient in solving each subtask.

Identification of inter-clausal temporal relations between the central events of two clauses involved in a syntactic dependency relation

The data for this task is obtained in a similar manner to the data for the previous task of intra-clausal event-event temporal relation annotation. Only the core set of three temporal relations is used. The scope of the temporal relations changes, as in this case only temporal relations between two clauses involved in a dependency relation are extracted.

In the following the focus will be on identifying the temporal relation holding between the central events of two clauses, provided that one clause is the temporal adjunct of the other clause. The methodology described in Section 7.2 is employed to identify temporal clauses introduced by ambiguous subordinators (those included in SAS), while the clauses introduced by non-ambiguous temporal connectives (those in STS \ SAS) are considered by default temporal clauses. To this end, the first step is extracting from TimeBank all possibly temporal clauses

introduced by any connective in STS together with their superordinate clauses. They are selected automatically so that the verb phrase of the subordinate clause is directly dependent on the verb phrase of the main clause and so that the relation between the two clauses (relation provided by the syntactic parser) is of adverbial nature (e.g. a subordinate clause that according to the syntactic parser is the direct object of the main clause is not considered).

The subordinate clauses corresponding to each ambiguous connective in SAS are manually annotated as temporal or non-temporal to be able to evaluate the performance of the machine learning algorithm described in Section 7.2.3 on this test data. The personalised classifiers presented in Section 7.2.3 are then trained on the data described in 7.2.2 and applied to the pairs of main-subordinate clauses extracted from TimeBank. According to the syntactic parser, TimeBank contains 65 subordinate adverbial clauses introduced by any of the following ambiguous subordinators: *when*, *as*, *while/whilst*, *since*, *as long as/so long as*. The accuracy of these personalised classifiers on the pairs of clauses extracted from TimeBank is: 92% for distinguishing between temporal and non-temporal clauses introduced by *when*, 81.81% for clauses introduced by *as*, 90.90% in the case of *while/whilst*, and 100% for *since*-clauses. According to the syntactic parser, there are no adverbial clauses introduced by *as long as* or *so long as* in the corpus.

A closer look at the TimeBank clause pairs reveals that out of the total 65 clauses introduced by ambiguous subordinators, 33 are temporal clauses, and the rest non-temporal. One possible baseline for the task of distinguishing between temporal and non-temporal clauses would be to consider all clauses temporal, this yielding an accuracy of 50.76%. The system using personalised classifiers for each ambiguous subordinator correctly classifies 58 clauses as temporal or non-temporal, thus achieving a score of 89.23% and bringing a

substantial improvement over the baseline. An important problem interfering with classifier performance appears due to errors made by the syntactic parser, either in linking the subordinate clause to the wrong main clause or to the wrong main clause constituent, or in wrongly identifying the verb phrases of the main and subordinate clauses.

When looking at the temporal relations between the temporal clauses and their superordinate clauses, one discovers that out of the 33 temporal clauses introduced by ambiguous subordinators, 12 are not linked via any temporal relation to the head of the main clause in the annotation present in TimeBank. Since this work does not focus on establishing which pairs of entities should be involved in a temporal relation, only the pairs of clauses linked by a temporal relation in TimeBank are further considered for system evaluation. Out of 21 clause pairs linked via a temporal relation in TimeBank, the system identifies the correct temporal relation using the methodology described in Section 7.3.2 in 17 cases, meaning that in 80.95% of the cases it identifies the correct temporal relation.

Besides clauses introduced by ambiguous subordinators, TimeBank also includes 29 temporal adverbial clauses introduced by non-ambiguous subordinators (those in STS \ SAS). In 12 cases, no temporal relation between the subordinate and main clause is annotated in TimeBank. Out of the 17 cases with a temporal relation associated in TimeBank, the system correctly specifies the temporal relation in 12 cases. The 5 system errors appear in the case of *until*-clauses, as the system always labels a temporal relation between the until-clause and the main clause with AFTER, while the human annotators annotated 3 of the cases with OVERLAP and 2 with BEFORE. While the cases annotated with OVERLAP are most probably annotation errors that could be avoided by

specifying clearer and more detailed annotation guidelines for smaller and more specific tasks, the cases annotated with BEFORE are due to negation and modal modification being present in the verb phrase of the main clause. The presence of negation probably motivated the annotators to reverse the temporal relation, as in the case of the example [7.26], where the event *warrant* is annotated as temporally before the event *resume*.

[7.26] *He said construction wouldn't <resume>* ***until*** *market conditions <warrant> it.*

The present work did not attempt to reverse temporal relations in such cases, as it is obvious that negated and modally subordinated events are marked using the attributes polarity and modality of the TimeML tag <EVENT>, and the temporal relation involves the fully modified event, and not only the markable alone as if it would not be marked for polarity and modality. This work assumes that the temporal relation between *warrant* and *would not resume* is AFTER, and the inferences derived from polarity and modality applied to any event involved in a temporal relation should be made by the temporal reasoner that takes the output of the TimeML annotation process and makes inferences on the basis of this annotation. However, such cases should be tackled in detail in the TimeML annotation guidelines, to avoid any misinterpretations and wrong annotations.

The overall system performance in identifying the temporal relation holding between a temporal clause and its matrix clause is 76.31% (29 correct out of 38 temporal relations between clause pairs linked via a temporal relation in TimeBank). A possible baseline would involve assigning the most frequent relation encountered in the data (BEFORE) to all clause pairs. This baseline would achieve a score of 55.26%.

7.4 Placing events in time with respect to the Document Creation Time

In a similar manner to identifying intra-sentential temporal relations, the system can perform the identification of temporal relations between any event and the DCT.

The processing stages for solving this task follow the course of the ones presented in Figure 7.2, with the only difference that the inter-clause and intra-clause temporal ordering modules no longer order clauses/constituents with respect to each other and in a bottom-up manner, but with respect to the DCT going top-down through the syntactic tree and employing the knowledge gained as a result of identifying intra-sentential temporal relations, knowledge concerning the relative ordering between same clause constituents.

In establishing a temporal relation between an event and the DCT, the temporal expressions directly or indirectly linked to that event are first analysed and, if no relation is detected, the temporal relation with the DCT is propagated top-down in the syntactic tree using the father node's temporal relation with the DCT and the temporal relation between the two constituents. In the case of any clause verb phrase, the relation with the DCT is found on the basis of the VP tense, the superordinate clause's VP tense, the syntactic relation connecting the clause with its superordinate and the relation between the superordinate clause's VP and the DCT.

7.4.1 Evaluation

The system capability to place events in time with respect to the DCT was evaluated in the context of the TempEval campaign. Table 7.6 presents the

Event-DCT temporal ordering	STRICT SCORE			RELAXED SCORE		
	P	R	F	P	R	F
BASELINE	0.62	0.62	**0.62**	0.62	0.62	**0.62**
TempEval-TRAIN	0.80	0.80	**0.80**	0.81	0.81	**0.81**
TempEval-TEST	0.80	0.80	**0.80**	0.80	0.80	**0.80**
TimeBank	0.80	0.80	**0.80**	0.81	0.81	**0.81**

Table 7.6: System results for Event-DCT temporal relation detection

system's evaluation results on the TempEval training data, TempEval test data, as well as the entire TimeBank corpus, along with a baseline. The baseline is established by the most frequent temporal relation encountered in the training data for temporal relations between events and the DCT, this relation being BEFORE.

The system presented in this thesis achieves high results in the discovery of temporal relations between events and the DCT, results substantially above the baseline (18%) and above the results achieved by any other system at TempEval both in the strict and relaxed settings. The official results of all the systems that participated in this task can be found in Table 7.7.

TEAM	STRICT SCORE			RELAXED SCORE		
	P	R	F	P	R	F
CU-TMP	0.75	0.75	0.75	0.76	0.76	0.76
LCC-TE	0.75	0.71	0.73	0.76	0.72	0.74
NAIST	0.75	0.75	0.75	0.76	0.76	0.76
USFD	0.73	0.73	0.73	0.74	0.74	0.74
WVALI	**0.80**	**0.80**	**0.80**	**0.81**	**0.81**	**0.81**
XRCE-T	0.78	0.57	0.66	0.84	0.62	0.71

Table 7.7: Official TempEval results for ordering events with respect to the DCT

7.5 Identification of inter-sentential temporal relations

Another interesting problem when trying to temporally order events is finding the temporal relation between events situated in different sentences. Due to the high complexity of this problem, TempEval proposes a task that represents an initial attempt of going beyond sentence level when temporally ordering events. It reduces the problem of detecting inter-sentence temporal relations to the task of relating the main events of two adjacent sentences. The main event of a sentence is considered to be the syntactically dominant verb of that sentence.

The approach taken in this work towards the identification of inter-sentential temporal relations initially relies on several heuristics (36) that involve the temporal expressions and the tensed main verbs of the two sentences to be temporally related. If no temporal relation can be inferred on the basis of these heuristics, the system then uses statistical data extracted from the TimeBank corpus that captures the most frequent temporal relation between two tensed verbs characterised by their tense and aspect.

Figure 7.4 illustrates the processing flow involved in temporally ordering the pair of events signalled by the main verbs of two consecutive sentences.

The two sentences are first parsed using Connexor's FDG parser and then clause boundaries are identified. The next step is locating the central verb of the main clause for each of the two sentences.

All TEs situated in the same clause with each main verb are investigated to see if these TEs and the relations between them and the two main verbs are able to predict a temporal relation.

In case no relation can be predicted, the next stage is investigating the

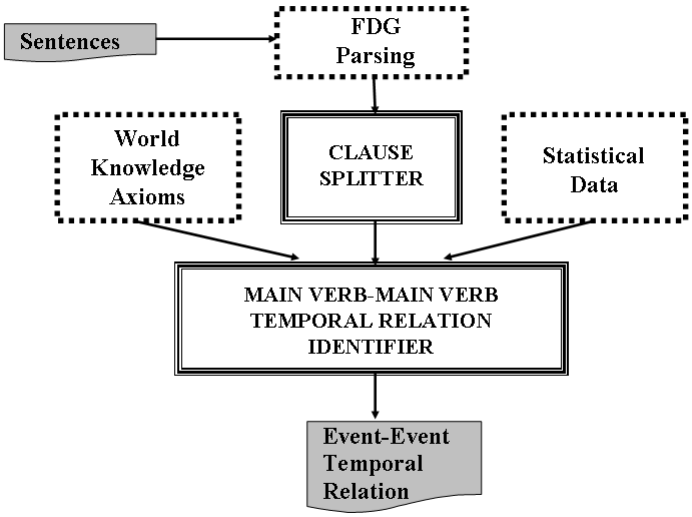

Figure 7.4: Processing stages for the inter-sentential temporal relation identifier

semantic properties of the two main verbs to detect whether they denote reporting events or not.

If both main verbs are reporting events then their tense information is used to predict a relation.

If only one main verb is a reporting event, then the TEs linked to the other main verb, if they exist, are used to infer a relation between the second main verb and the DCT. The assumption is that a reporting event is located temporally simultaneous with the DCT and, if a relation between the second event and the DCT can be established by means of surrounding TEs, then this is the relation providing the system output. If the non-reporting event can not be positioned in time with respect to the DCT by analysing surrounding TEs, then its relation with the DCT will be the one established as described in Section 7.4.

The most complicated case is when both main verbs are non-reporting events. This case is solved by picking for each tense pair the most frequent temporal relation in the corpus, unless there is a tie or another relation with

Temporal Relation	Temporal Relation	Reconciled Relation
OVERLAP	BEFORE-OR-OVERLAP	BEFORE-OR-OVERLAP
OVERLAP	BEFORE	BEFORE-OR-OVERLAP
OVERLAP	OVERLAP-OR-AFTER	OVERLAP-OR-AFTER
OVERLAP	AFTER	OVERLAP-OR-AFTER
BEFORE	BEFORE-OR-OVERLAP	BEFORE-OR-OVERLAP
AFTER	OVERLAP-OR-AFTER	OVERLAP-OR-AFTER
VAGUE	any relation	any relation

Table 7.8: Reconciliation between temporal relations for inter-sentential temporal ordering

very similar frequency occurs, in which cases the two temporal relations are reconciled according to Table 7.8. To detect whether the first two most frequent temporal relations need to be reconciled, the percentage distribution of all possible temporal relations associated with a given tense pair is calculated. Then the percentages corresponding to the two most frequent temporal relations associated to that tense pair are compared, and they are considered to be very similar when the difference between them is lower than a threshold of 5%, case in which they are reconciled. In this manner a temporal relation is associated with each tense pair and, consequently, the temporal relation between the two main verbs is identified.

7.5.1 Evaluation

The system's capability to order the main events of two consecutive sentences was evaluated in the context of the TempEval campaign. Table 7.9 presents the system's evaluation results on the TempEval training data, TempEval test data, as well as the entire TimeBank corpus, along with a baseline. The baseline involves assigning to all event pairs the most frequent temporal relation

Inter-sentence temporal ordering	STRICT SCORE			RELAXED SCORE		
	P	R	F	P	R	F
BASELINE	0.42	0.42	**0.42**	0.46	0.46	**0.46**
TempEval-TRAIN	0.53	0.53	**0.53**	0.63	0.63	**0.63**
TempEval-TEST	0.54	0.54	**0.54**	0.64	0.64	**0.64**
TimeBank	0.53	0.53	**0.53**	0.63	0.63	**0.63**

Table 7.9: System results for inter-sentential temporal ordering

encountered in the training data for this task, this relation being OVERLAP.

Despite the challenges posed by inter-sentential temporal relation identification and of the simplistic approach taken in this work for their identification, the present system achieved the best relaxed score among all participants at TempEval. The official results of all the systems that offered a solution to this task can be found in Table 7.10.

TEAM	STRICT SCORE			RELAXED SCORE		
	P	R	F	P	R	F
CU-TMP	0.54	0.54	0.54	0.58	0.58	0.58
LCC-TE	0.55	0.55	0.55	0.58	0.58	0.58
NAIST	0.49	0.49	0.49	0.53	0.53	0.53
USFD	0.54	0.54	0.54	0.57	0.57	0.57
WVALI	**0.54**	**0.54**	**0.54**	**0.64**	**0.64**	**0.64**
XRCE-T	0.42	0.42	0.42	0.58	0.58	0.58

Table 7.10: Official TempEval results for ordering the main events of two consecutive sentences

While the other two temporal relation identification tasks can be solved with satisfactory results using mostly syntactic information and very little semantic information, finding a solution to the problem of inter-sentential temporal ordering is heavily reliant upon semantic information. For a better system performance, one needs access to different types of semantic information, such as causality relations, event part-whole relations, and textual entailment information.

7.6 Conclusions

This chapter focused on the identification of temporal relations that can be established among events and temporal expressions. Language uses a variety of mechanisms to express temporal relations, the most frequent including tense, aspect and time adverbials. Tense and aspect, as two important attributes characterising events, were identified as part of the event annotation process (please refer to Chapter 6 for more details). Time adverbials expressed using adverbial phrases, noun phrases and prepositional phrases were identified and normalised using the methodology described in Chapters 4 and 5). Temporal clauses are another important subclass of time adverbials, and their identification was addressed at the beginning of this chapter in Section 7.2.

A machine learning approach for the identification of temporal clauses was proposed. A classifier was trained for each temporal connective manifesting semantic ambiguity, and their performance in distinguishing between a connective's temporal and non-temporal usages ranged from 82.14% to 98.33%. Their accuracy is very good considering that they rely mainly on surface and syntactic features, but it can definitely be improved by adding semantic information mainly about the verbs occurring in the main and subordinate clauses.

A novel methodology for the identification of temporal relations was devised following careful examination of the mechanisms used by language to express temporal relations and relying on previously implemented system capabilities to identify them. This methodology was specifically designed to automatically identify the temporal relations that hold between any two temporal entities situated in the same sentence, between any event and the speech time (represented

by the Document Creation Time in the case of news articles), as well as between two main events located in two consecutive sentences.

The novel approach for discovering intra-sentential temporal relations relies on sentence-level syntactic trees and on a bottom-up propagation of the temporal relations between syntactic constituents, by employing syntactical and lexical properties of the constituents and the relations between them. A temporal inference mechanism is afterwards employed to relate any two targeted temporal entities to their closest ancestor and then to each other. Using this approach, one can discover temporal relations between any two events or between any event and any TE, whenever the two entities are situated in the same sentence. The task of linking two temporal entities situated in the same sentence has been addressed by the TempEval evaluation exercise. Low human annotator agreement together with lessons learned from participating in TempEval have led to designing syntactically motivated subtasks and addressing them in this work. Evaluation results have shown that these subtasks can be reliably resolved, leading to the conclusion that it is highly advisable to decompose the temporal relation annotation task into smaller well-defined subtasks. After finding accurate and linguistically grounded solutions to these smaller subtasks, one can then proceed to a higher level by composing more complex tasks once gaining knowledge as to how different temporal phenomena interact when they are grouped in complex utterances.

Another problem dealt with in this work is the identification of temporal relations between any event and the DCT. In establishing a temporal relation between an event and the DCT, the temporal expressions directly or indirectly linked to that event are first analysed and, if no relation is detected, the temporal relation with the DCT was propagated top-down in the syntactic tree.

The problem of inter-sentential temporal ordering is reduced to identifying the temporal relation between the main events represented by the verbs heading the syntactic trees of the two sentences. Inter-sentence temporal relations are discovered by first applying several heuristics that involve the temporal expressions and the tensed verbs corresponding to the main clauses of the two sentences to be temporally related, and then by using statistical data extracted from the TimeBank corpus that provides the most frequent temporal relation between two tensed verbs characterised by tense information.

The solutions to these three problems, along with evaluation results were presented in detail in Sections 7.3 to 7.5.

The main advantage of the approach proposed in this work is the fact that the architecture and core modules are domain independent, since they mainly rely on generic correlations between syntax and temporality. This approach is domain independent and can be easily adapted to a new domain as long as the analysed texts are syntactically correct. At a change of domain, only the heuristics involving the DCT and the reporting events implicitly located on the date of the article need to be eliminated. Obviously for each domain certain domain-dependent rules can improve the system's accuracy on texts belonging to that domain, but the core approach remains unchanged.

The system implementing the methodology described in this chapter has been tested and evaluated within the framework established by the TempEval evaluation exercise organised as part of SemEval-2007, where it achieved the best results among all participating systems. One can therefore conclude that the proposed approach is appropriate for discovering temporal relations.

Chapter 8

Conclusions

8.1 General conclusions

This thesis focused on the investigation and understanding of the different ways time is expressed in natural language, on the implementation of a temporal processing system inspired from the results of this investigation, on the evaluation of the system, and on the extensive analysis of the errors and challenges that appeared during system development.

The work presented in this thesis is not a piece of research in linguistics, but in language engineering. Therefore, the main stress was on the implementation of a practical system that relies on linguistic theories, on its efficiency and effectiveness. The main requirements of the system were not only to achieve good performance, but also to be fast, robust and reliable, and, last but not least, to be modular enough to enable its integration in a larger NLP application.

In designing the methodology and the system implementing it, the main aim was to make them as general-purpose as possible, to ensure their versatility and wide applicability. Even if the methods involved in the annotation of different types of temporal information were developed, tested and evaluated on news articles, there are no rules that are specific to a certain genre or domain to such

an extent as to make the methodology inapplicable to other types of text. The choice of news articles was based on very practical considerations, related mostly to the availability of previously annotated data.

In light of all this, the main contributions of this thesis include:

A novel methodology for the identification and annotation of different types of temporal information in text

This thesis addressed the automatic identification and annotation of the following types of temporal information: temporal expressions, verbal events, and the temporal relations holding among them. The original contributions brought by this work to the automatic treatment of each temporal information type are illustrated below.

- **Temporal expressions**

  Temporal expressions can be of various types, and any system targeting their annotation needs to be aware of these types and provide each TE the appropriate treatment in line with the semantic properties characterising its type. This work includes an exhaustive classification of TEs, due to its usefulness at several system development stages. Despite the fact that various distinctions between TE types have been previously mentioned by other researchers, this is the first time a clear and detailed classification of TEs has been published.

  The TE identification and normalisation modules are developed on the basis of this classification. At the identification stage, finite state automata first pinpoint sequences of words corresponding to possible TEs, then the identified

TEs are checked for syntactic correctness and transformed into well-formed syntactic constituents.

The problem of *then*, one of the most frequent English temporal adverbials, cannot however be solved syntactically. The fact that *then* can express different semantic roles, and only those usages that realise the semantic role of time were to be annotated as temporal expressions, required further attention. The disambiguation of *then* was first approached by annotating a corpus capturing its different usages, and training a machine learning classifier on this data. An empirical approach was then developed on the basis of a rigorous linguistic investigation of *then*. Both approaches achieved good results, and they are both unique in the specialised literature, as the adverb *then* has so far only been investigated from a theoretical perspective.

The TE identification stage is normally followed by normalisation, the process that assigns to each identified TE a series of attributes and attribute values in accordance with a chosen annotation scheme. At this level, this work brings an original contribution by investigating the impact of different temporal anchor tracking models on the overall normalisation process. Existing approaches used the document timestamp to calculate the value designated by an underspecified TE. This thesis proposed four temporal anchor tracking models, all having different levels of context dependency, and relying on the distinctions present in the TE classification.

The work devoted to temporal expressions has initially targeted the TIMEX2 annotation standard, due to its high level of refinement and reliability among temporal annotation schemes. Given the aim of this thesis to cover the three main classes of temporal entities, and that the worldwide adopted

standard for their annotation is TimeML (ISO-TimeML, 2007), the system developed for TIMEX2 annotation was enhanced with capabilities to perform TimeML/TIMEX3-compliant TE annotation. The adaptation process and its detailed description are a novel contribution to research in temporal processing.

- **Verbal events**

  The identification of verbal events can be done reliably using information provided by the syntactic parser. The annotation of verbal events with TimeML-compliant information requires the assignment of an aspectual class to every event. Determining the right aspectual class that should be assigned to an event is the biggest problem of verbal event annotation systems. This problem is solved in this thesis via an annotation process targeting all the verbs present in WordNet 2.0, and assigning to each verb its most relevant event class.

  The method typically employed by other researchers in classifying events into event classes was very rudimentary, tagging events with the class that was most frequently assigned to them in TimeBank. Other approaches described in detail in Section 3.5 trained different classifiers for distinguishing between event classes, mainly by looking at co-occurrence patterns that were similar to events annotated in TimeBank. The first method performed well only for words annotated in TimeBank and it could not cater for verbs not included in TimeBank. The second method tried to overcome TimeBank's limits but was still highly dependent on TimeBank and its performance was around 60%, thus not ensuring reliability. This research and the methodology proposed here bring a novel contribution to this area by offering the research community a reliable method to identify and classify verbal events in any natural language text, irrespective of their appearance in TimeBank. Unlike

previous approaches, this method does not depend on the information annotated in TimeBank, information that can be sometimes unreliable due to many annotation inconsistencies.

- **Temporal relations**

  Among the mechanisms used by language to express temporal relations, tense, aspect and time adverbials are extremely important, as they provide explicit information that is automatically identifiable and can be exploited in the development of systems targeting the identification of temporal relations. Tense, aspect and most time adverbials were automatically identified as part of the event and temporal expression annotation process. A subclass of time adverbials that has received little attention from researchers is represented by temporal clauses. This work addressed this shortcoming, and proposed a methodology for identifying temporal clauses in text by adopting a machine learning method that detects when ambiguous subordinators are used to introduce temporal clauses. The clauses introduced by unambiguous temporal subordinators were considered temporal clauses by default.

  The system capabilities to identify these mechanisms were then exploited in modules designed to automatically identify the temporal relations that hold between any two temporal entities situated in the same sentence, between any event and the Document Creation Time, as well as between two main events of two consecutive sentences.

  Investigations have shown that most temporal relations marked by human annotators link temporal entities situated in the same sentence (more details can be found in Section 7.3). This justifies the number of experiments and the

attention dedicated to intra-sentential temporal relations in this thesis. Intra-sentential temporal relations were identified in this work by relying on sentence-level syntactic trees and a bottom-up propagation of the temporal relations between syntactic constituents, by analysing syntactic and lexical properties of the constituents and of the relations between them. Given any two temporal entities situated in the same sentence, a temporal inference mechanism was afterwards employed to relate each of the two entities to their closest ancestor and then to each other. This method achieves good results when compared to other systems performing the same task. However, several factors such as the low level of agreement observed between human annotators, and the evaluation results obtained by systems that participated in the TempEval campaigns, have proved that the task has been incorrectly defined, in the sense that not any pair of co-sentential entities should be necessarily involved in a temporal relation.

The need for task decomposition, and for designing syntactically motivated subtasks has been generally consented to. This work addressed this need by defining four subtasks according to different syntactic criteria. Linguistically informed solutions have been detailed for each subtask, the evaluation results showing that these subtasks can be reliably resolved automatically. The conclusion was that it is highly advisable to decompose the temporal relation annotation task into smaller well-defined subtasks, and that after finding accurate and linguistically grounded solutions to these smaller subtasks, one can then proceed to a higher level by composing more complex tasks. The tasks should be defined in an order that reflects increasing amounts of context and increasing degrees of difficulty. The work on task decomposition presented in this thesis is novel and has not been addressed before by other researchers.

Another contribution of this thesis is a novel methodology to identify temporal relations between any event and the Document Creation Time. When establishing the temporal relation between an event and the DCT, the temporal expressions directly or indirectly linked to that event were first analysed and, if no relation was detected, the temporal relation with the DCT was propagated top-down in the syntactic tree.

The problem of finding the temporal relation between the main events of two consecutive sentences was solved using the temporal expressions and the tensed verbs corresponding to the two main event clauses, and when this information proved to be inconclusive, statistical data extracted from the TimeBank corpus helped decide the temporal relation.

An extensive comparative evaluation and error analysis

An important part of this research has been the evaluation and error analysis of the methods proposed for solving different temporal processing tasks. The contributions brought by this work in this area are presented below for each type of temporal information tackled.

- **Temporal expressions**

  In the case of the temporal expression identification task, detailed comparative evaluation results were obtained by decoupling the subtasks involved and illustrating the improvement in performance obtained after each processing stage. Evaluation results showed that the module that checks for syntactic correctness brought a statistically significant improvement in system performance. The results continued to improve after adding the module dealing

with the disambiguation of *then*, but the improvement was not statistically significant due to the low frequency of *then* in the evaluation corpus. The system performance for the annotation of TIMEX2 temporal expressions was 95.30% for partial matches, and 86.30% for exact matches. For TIMEX3 annotation, the system achieved 91.80% for partial matches, and 86.70% for exact matches. In the case of TIMEX3 annotation, no other comprehensive evaluation has been provided in the literature, most systems being evaluated only for the task of TIMEX2 annotation. A detailed analysis of the errors that appeared at the TE identification stage revealed several error sources including: syntactic parser errors, patterns that were not implemented, lexical triggers that were not present in the lexicon, as well as errors involving legitimate TEs that were missing or wrongly annotated in the gold standard. It is interesting to see that 29.55% of the errors that appeared during the TIMEX2 annotation process were human annotator errors, while in the case of the TIMEX3 annotation process 77.78% of the errors were made by human annotators. This result stands as proof for the fact that the TIMEX3 annotation guidelines and the TimeBank corpus still require improvement to reach the level of detail and reliability of the TIMEX2 annotation guidelines and of the TERN corpus.

At the normalisation stage, four temporal anchor tracking models have been evaluated in turn to discover the best approach for identifying the anchor of an under-specified TE. A comparative evaluation of these models has shown that the best performing temporal anchor tracking model was the class-sensitive normalisation model prioritising clause-local context. The system implementing this model was then further enhanced with a module that addressed the direction problem, and then with a module dealing with the generic vs. specific problem, and evaluations were performed after each

addition. Both modules improved the system, the final accuracy being 88% for assigning values to the VAL attribute. Detailed error analysis was performed at each system development level. After adapting the system from TIMEX2 to TIMEX3 annotation, another evaluation is performed, followed by a detailed error analysis that facilitates a comparison between the types of errors that appear in the TIMEX2 and TIMEX3 normalisation processes.

- **Verbal events**

  The identification and annotation of verbal events according to the TimeML standard have been evaluated in this work as separate tasks. The evaluation and error analysis have focused on these two tasks addressed from the perspective of verbal events expressed using finite verbs and non-finite verbs.

  The accuracy of the system was 86.68% for identifying finite verb events, 73.45% for identifying non-finite verb events, and 86.49% for the overall identification of verbal events. Errors were mainly caused by over-annotation, in the sense that the system identified more events than those present in the corpus. This was due to several reasons. On the one side, the system failed because of errors introduced by the syntactic parser, and also because it was not able to deal with generic verb mentions which, according to the guidelines, should not be considered events, and the system annotated them as events. On the other side, many verb occurrences were missed by the human annotators who should have annotated them. Such discrepancies account for 39.01% of the errors observed during finite verb event identification, and for 54.42% of the errors corresponding to non-finite verbal events. These figures confirm the hypothesis that finite verbs capture the most important information in a sentence, and therefore humans focus more on them when annotating

events, while often considering the information expressed by non-finite events not relevant for event annotation purposes.

The system accuracy obtained when assigning values to the TimeML attributes associated with an event ranges from 85.57% for the attribute **class**, to 94.60% for **tense**, 98.19% for **aspect**, 99.08% for **polarity**, and 99.28% for **modality**. The result of 85.57% in the case of the **class** attribute is a very good result when considering that the approach taken in this work assigned one class per verb.

- **Temporal relations**

  This thesis mainly focused on the identification of temporal relations between temporal entities situated in the same sentence, but it also proposed methods for finding the temporal relations holding between an event and the Document Creation Time, and between the main events of two consecutive sentences. The system solving these tasks has been evaluated in the context of the TempEval evaluation exercise. Following lessons learned from TempEval, the task of identifying intra-sentential temporal relations has been decomposed into smaller subtasks motivated syntactically. Some of these subtasks have been identified and evaluated in this work. For example, the accuracy for the identification of intra-clausal temporal relations between a TE and a governing verbal event was 84.80%, substantially higher than the f-measure of 62% observed during TempEval when all possible pairs of co-sentential entities had to be linked with a temporal relation. The evaluation of these subtasks has shown that they can be reliably resolved automatically, leading to the conclusion that it would be highly advisable to decompose the temporal relation annotation task into smaller well-defined subtasks. After designing linguistically grounded solutions for solving

these smaller subtasks accurately, more complex tasks could be devised and solved by gaining knowledge about how different temporal phenomena interact when they are grouped in complex utterances.

Development of novel resources

The work presented in this thesis involved the development of several corpora annotated for various purposes. The annotation of each corpus was useful both for analysing a certain temporal phenomenon, and for training and testing methods that automatically deal with that phenomenon. These are reusable resources that are an important contribution to the research community. The resources developed as part of this work are presented below.

- **A corpus illustrating different usages of *then***

  This corpus was used for training and testing the methods employed for the disambiguation of *then*. The annotated data contains 1,173 occurrences of *then*.

- **A resource that associates with each verb present in WordNet 2.0 the event class that best characterises that verb**

  This resource maps each of the 11,306 verbs present in WordNet 2.0 to an aspectual event class that best captures that verb's meanings. It has been used in this work as part of the event annotation process to provide the value of the attribute **class** included in the TimeML EVENT tag.

- **A corpus capturing the behaviour of ambiguous subordinators that can be used to introduce temporal clauses**

  In this data collection, each subordinator was assigned a class that delimited

a temporal usage from a non-temporal one. This corpus was used for training and testing classifiers with the aim of identifying temporal clauses.

8.2 Research goals revisited

This section illustrates how the goals outlined in the introductory chapter have been achieved in this thesis.

Goal 1 was to review how temporal information is conveyed in natural language. Chapter 2 described from a theoretical perspective the most important mechanisms used by language to express temporal information, thus addressing this goal.

Goal 2 was to overview existing approaches in automatic temporal processing. Chapter 3 accomplished this goal by presenting temporal annotation schemes, resources, and computational approaches employed so far to perform different temporal processing tasks.

Goal 3 was to develop the corpora required to investigate different phenomena that needed to be tackled in this research. Several chapters contributed to addressing this goal. Chapter 4 describes the corpus that captures the different semantic categories expressed by *then*. Chapter 6 presents the annotation process of the resource that associates each WordNet verb with its aspectual class. Chapter 7 includes the description of the corpus built with the purpose of identifying temporal clauses introduced by ambiguous subordinators.

Goal 4 was to design, implement and evaluate the methodology concerned with temporal expression identification. These objectives were achieved in Chapter 4 which focused on the processing stages involved in identifying the textual extent of temporal expressions, on their comparative evaluation, and on their adaptation to a different annotation standard.

Goal 5 was to investigate how each occurrence of the temporal adverb *then* could be automatically disambiguated to distinguish the anaphoric occurrences of *then* that act as temporal expressions and require annotation accordingly. Chapter 4 fulfilled this goal by proposing a machine learning and an empirical approach for the disambiguation of *then*.

Goal 6 was to identify the best approach to be adopted when normalising temporal expressions. Chapter 5 presented several normalisation models and identified the most important problems that appear during normalisation. The solutions proposed and their comparative evaluation revealed the best approach to be adopted, thus accomplishing this goal.

Goal 7 was to design and evaluate a method for the identification and annotation of verbal events in text. Chapter 6 achieved this goal by describing the annotation process of a resource that associated each verb with an aspectual class, and the way finite and non-finite verbal events were identified and annotated with their corresponding TimeML attributes.

Goal 8 was to propose a methodology for the identification of temporal relations holding between events and temporal expressions, or between events and

other events. This goal was accomplished in Chapter 7, where a novel methodology for the identification of temporal relations was investigated and evaluated in different settings.

Goal 9 was to find a way to automatically identify temporal clauses in text. This problem was solved in Chapter 7 with a machine learning approach that was able to decide whether a clause was temporal or not by disambiguating the subordinator that introduced that clause, if that subordinator was known to be ambiguous.

Goal 10 was to identify limitations of this work, and to identify future directions of research. This goal is addressed in this chapter.

8.3 General overview of the thesis

This section provides a general overview of the thesis by summarising each chapter.

Chapter 1 presented an introduction to this research by capturing the motivations behind studying this particular topic that lie in the possible applications of temporal processing in NLP, the original contributions made by this work, and the goals that were set to be achieved in this thesis.

Chapter 2 looked at how time is expressed in natural language, and described the different types of temporal information and the linguistic efforts made to formalise them. This chapter also included a survey of previous work that focused on the theoretical aspect of temporal processing.

Chapter 3 performed a comprehensive literature review from a practical perspective. It presented existing annotation schemes, resources, and computational approaches previously adopted for solving temporal processing tasks. The critical analysis of previous work has helped identify the best course to follow in this work.

Chapter 4 described the methodology adopted in this research to address the task of temporal expression identification. This chapter offered solutions to the problems that appeared in the process, presented detailed evaluation results and error analysis, and demonstrated that the methodology and representation chosen in this work were general enough to facilitate adaptation to a different annotation scheme than the one initially adopted.

Chapter 5 focused on the task of temporal expression normalisation, and proposed several alternatives for selecting the anchor that contributed to resolving an under-specified TE. The problems that appeared during normalisation were also addressed in this chapter. The influence of each module involved in the normalisation task on the overall system performance was evaluated, and the best normalisation setting was then adapted to the TIMEX3/TimeML annotation scheme, followed by another evaluation.

Chapter 6 presented the methodology adopted in this work for the identification and annotation of events denoted by verbs. The event identification process relied on information provided by the syntactic parser, while the event annotation process required not only syntactic information, but also access to the event's semantics. Determining the aspectual class of an identified event is the most complicated part of the annotation process due to its semantic nature. This

problem was solved in this thesis by annotating each WordNet verb with the most suitable aspectual class for that verb's meanings. The resulting resource was then used to associate an aspectual class with each identified verbal event. Detailed evaluation results and error analysis were also included in this chapter.

Chapter 7 proposed a new methodology for identifying temporal relations that hold among events and temporal expressions, a methodology that was devised following careful examination of the mechanisms used by language to express temporal relations. Various settings were evaluated in this chapter, and their results and problems encountered were presented in detail. Most of the evaluation results reported in this chapter were obtained in an independent setting offered by the TempEval evaluation exercise, where the system described in this chapter achieved the best performance.

8.4 Future research directions

Further work stemming from this research involves specific tasks that would improve the functionality of the system described in this thesis, or wider applications that would use this system to address more complex NLP problems.

When considering the first category, one possible line of research would be to adapt and evaluate the system on texts belonging to other genres. In this work, the system processed only news articles, mainly due to them being the only available resource annotated with temporal information. Therefore, the ability to follow this line of research is directly dependent on the availability of temporally annotated data from different genres.

One specific task that could be investigated further is the temporal expression

normalisation task. Chapter 5 examined several normalisation models, all having different levels of context dependency and awareness of surrounding discourse. Evaluation results have shown that the best normalisation results were achieved by the model that imposed limitations on the domain of referential accessibility of an under-specified TE by prioritising clause local context. It would be interesting to see how discourse structure influences the choice of a temporal anchor by examining various discourse theories with the aim of identifying the most salient temporal expressions mentioned in the discourse that precedes the TE to be normalised. Such an inquest would help develop temporal normalisation models characterised by enhanced discourse awareness, and possibly by better performance in locating the temporal anchor required for resolving an under-specified TE. However, this type of investigation requires a corpus annotated with explicit information about the anchor used by the human annotator to calculate the final value associated to the under-specified TE.

The task of nominal event recognition is an extremely relevant topic for future research. This thesis has only focused on verbal events, but events can also be expressed using nouns, and one needs to find ways to identify them. An investigation of this problem is currently in progress. It relies on a bootstrapping technique and on patterns that trigger the extraction of a nominal event. Considerable effort needs to be invested in finding the appropriate filtering and ranking method that would guarantee a qualitative list of events as output.

Temporal relation identification is another research area that could be pursued further. An essential stage in finding the temporal relation between two temporal entities is detecting the temporal relation that holds between a pair of clauses involved in a syntactic relation. Existing connections between syntax and temporality need to be further investigated at inter-clausal level. For each type

of syntactic relation that can hold between two clauses, it would be interesting to extract from a corpus pairs of clauses involved in that relation, and to analyse the correlations that can be identified between the syntactic properties of the two clauses combined with the syntactic relation holding between them and the temporal relation that can be established between the main events of the two clauses. This analysis could suggest improvements to the module that solves the task of inter-clausal temporal ordering. It could also indicate other clear and focused sub-tasks that can be used in system evaluation than the ones evaluated in this work. This leads to another line of research involving not only the methodology to resolve these smaller sub-tasks, but also formalising the manner in which they interact and form more complex tasks. They should be defined, resolved and combined in an order that reflects increasing amounts of context and increasing degrees of difficulty.

When looking at wider applications that would rely on the temporal processing capabilities developed in this work, the possibilities are endless.

One research direction that would benefit the research community focusing on temporal processing is to design a computer-aided annotation process that would use the system developed in this work to assist human annotators in their work. By using the system to provide an automatic pre-annotation, the annotator only has to check and modify the system output, as opposed to creating everything from scratch. This would reduce the time needed for annotation, decrease the rate of annotation errors, and increase the efficiency of the annotation process. In this context, the system would benefit from a mechanism that assigns confidence values to its annotations, which is another problem that needs to be explored in the future.

Another line of research that could be pursued having this system as a starting point would be to integrate temporal processing in a larger application such as Question Answering. To this end, the system capabilities developed in this work would have to be applied at all the stages involved in the QA process, i.e. Question Processing, Paragraph Retrieval, and Answer Extraction. The methodology that would guide the integration of temporal processing in a Question Answering system offers a long-term research direction.

Appendix A

Previously published work

Some of the work described in this thesis has been previously published in proceedings of peer-reviewed international conferences. Before its inclusion in this thesis, most of this work has been extended or modified to address shortcomings and new research directions identified after the articles were published. This appendix provides a short description of these papers and explains their contribution to this thesis:

- **Georgiana Puşcaşu** (2004) "A Framework for Temporal Resolution". In *Proceedings of the 4th Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, May, pages 1901–1904

  This article proposes a framework for the identification and normalisation of temporal expressions in natural language texts. The work described in this article has been augmented with additional system capabilities to yield the temporal expression identifier and normaliser presented in Chapters 4 and 5.

- **Georgiana Puşcaşu** and Ruslan Mitkov (2006) "If *it* were *then*, then when was *it*? Establishing the anaphoric role of *then*". In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy, May, pages 1194–1199

This paper focuses on the disambiguation of the temporal adverb *then*. The machine learning method adopted is also presented in this thesis in Section 4.3.3.

- **Georgiana Puşcaşu**, Patricio Martinez Barco, and Estela Saquete Boro (2006) "On the Identification of Temporal Clauses". In *Proceedings of the 5th Mexican International Conference on Artificial Intelligence (MICAI 2006)*, Apizaco, Mexico, November, pages 911–921

  This article presents a machine learning approach to the identification of temporal clauses by disambiguating the subordinating conjunctions used to introduce them. This approach is used with very few modifications in this thesis, and forms the focus of Section 7.2.

- **Georgiana Puşcaşu** (2007) "WVALI: Temporal Relation Identification by Syntactico-Semantic Analysis". In *Proceedings of the 4th International Workshop on Semantic Evaluation (SemEval-2007) at ACL 2007*, Prague, Czech Republic, June, pages 484–487

  This paper describes the participation of the temporal relation identification system described in Chapter 7 in the TempEval evaluation campaign. In this thesis, a slightly modified and improved version of the algorithm is evaluated on several subtasks that represent refinements of the original TempEval tasks.

- **Georgiana Puşcaşu** (2007) "Discovering Temporal Relations with TicTac". In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2007)*, Borovets, Bulgaria, September, pages 493–498

  This article focuses on the methodology employed in this work to address the

problem of temporal relation identification at different levels: intra-sententially, inter-sententially, and with respect to the Document Creation Time. The methodology is illustrated in more detail in Sections 7.3, 7.4, and 7.5.

- **Georgiana Puşcaşu** and Verginica Barbu Mititelu (2008) "Annotation of WordNet Verbs with TimeML Event Classes". In *Proceedings of the 6th Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, May, pages 2793–2800

  This paper reports on the annotation of all English verbs included in WordNet 2.0 with TimeML event classes, and on the process that employs the resulted resource to automatically assign the corresponding class to each occurrence of a finite or non-finite verb in a given text. The efforts described in this paper are essential to the event annotation process described in Chapter 6.

References

Steven Abney. Partial Parsing via Finite-State Cascades. In *Journal of Natural Language Engineering*, volume 2, pages 337–344. 1996.

ACE. Automatic Content Extraction Evaluation, 1999. URL `http://www.itl.nist.gov/iad/mig//tests/ace/`.

ACL-2001. Workshop on Temporal and Spatial Information Processing, 2001. URL `http://www.aclweb.org/anthology-new/W/W01/#1300`.

Sigurd Agrell. *Aspektänderung und Aktionsartbildung beim polnischen Zeitworte: ein Beitrag zum Studium der indogermanischen Präverbia und ihrer Bedeutungsfunktionen. [Aspectual change and Aktionsart construction in the Polish verb: a contribution to the study of the Indo-European preverbs and their meaning functions.].* Lunds Universiteits Arsskrift, 1908.

David Ahn, Sisay Fissaha Adafre, and Maarten de Rijke. Towards Task-Based Temporal Extraction and Recognition. In *Proceedings of Dagstuhl Workshop on Annotating, Extracting and Reasoning about Time and Events*, Dagstuhl, Germany, 2005a.

David Ahn, Sisay Fissaha Adafre, and Maarten de Rijke. Extracting Temporal Information from Open Domain Text: A Comparative Exploration. In R. van

Zwol, editor, *Proceedings of the Fifth Dutch-Belgian Information Retrieval Workshop (DIR 2005)*, pages 3–10, 2005b.

David Ahn, Sisay Fissaha Adafre, and Maarten de Rijke. Extracting Temporal Information from Open Domain Text. In *Journal of Digital Information Management*, volume 3, pages 14–20. 2005c.

David Ahn, Sisay Fissaha Adafre, and Maarten de Rijke. Recognizing and Interpreting Temporal Expressions in Open Domain Texts. In *We Will Show Them: Essays in Honour of Dov Gabbay*, volume 1, pages 31–50, 2005d.

David Ahn, Joris van Rantwijk, and Maarten de Rijke. A Cascaded Machine Learning Approach to Interpreting Temporal Expressions. In *Proceedings of HLT-NAACL*, pages 420–427, 2007.

Jan Alexandersson, Norbert Reithinger, and Elisabeth Maier. Insights into the dialogue processing of VERBMOBIL. In *Proceedings of the fifth conference on Applied natural language processing*, pages 33–40, Washington, DC, 1997. Association for Computational Linguistics.

James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic Detection and Tracking Pilot Study, Final Report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, February 1998.

James Allan, Hubert Jin, Martin Rajman, Charles Wayne, Daniel Gildea, Victor Lavrenko, Rose Hoberman, and David Caputo. Topic-based Novelty Detection. In *Proceedings of the CLSP 1999 Summer Workshop*, 1999.

James Allen. Maintaining Knowledge about Temporal Intervals. In *Communications of the ACM*, volume 26, pages 832–843. November 1983.

James Allen. Towards a general theory of action and time. In *Artificial Intelligence*, volume 23, pages 123–154. Elsevier Science Publishers Ltd., July 1984.

James Allen. Time and time again: The many ways to represent time. In *International Journal of Intelligent Systems*, volume 6, pages 341–355. 1991.

Omar Alonso, Michael Gertz, and Ricardo Baeza-Yates. On the Value of Temporal Information in Information Retrieval. In *ACM SIGIR Forum*, volume 41. December 2007.

Ioannis Androutsopoulos. *Exploring Time, Tense & Aspect in Natural Language Database Interfaces*, volume 6 of *Natural Language Processing Series*. John Benjamins, 2002.

Ioannis Androutsopoulos. *A Principled Framework for Constructing Natural Language Interfaces to Temporal Databases*. PhD thesis, University of Edinburgh, 1996.

Ioannis Androutsopoulos, Graeme Ritchie, and Peter Thanisch. Time, Tense and Aspect in Natural Language Database Interfaces. In *Journal of Natural Language Engineering*, number 4, pages 229–276. 1998.

Chinatsu Aone, Lauren Halverson, Tom Hampton, and Mila Ramos-Santacruz. SRA: Description of the IE system used for MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1999.

Aristotle. Metaphysics, 350 BC. URL http://classics.mit.edu/Aristotle/metaphysics.html.

ARTE. ACL 2006 Workshop on Annotating and Reasoning about Time and Events, 2006. URL http://www.timeml.org/acl2006time/.

Emmon Bach. On time, tense and aspect. An essay in English metaphysics. In Peter Cole, editor, *Radical Pragmatics*, pages 63–81. Academic Press, New York, 1981.

Emmon Bach. The Algebra of Events. In *Linguistics and Philosophy*, volume 9, pages 497–508. 1986.

Adam Berger, Stephen Della Pietra, and Vincent Della Pietra. A Maximum Entropy Approach to Natural Language Processing. In *Computational Linguistics*, volume 22, pages 39–71. 1996.

Steven Bethard. *Finding Event, Temporal and Causal Structure in Text: A Machine Learning Approach*. PhD thesis, University of Colorado at Boulder, December 2007.

Steven Bethard and James H. Martin. Identification of Event Mentions and their Semantic Class. In *Proceedings of EMNLP'2006*, pages 146–154, 2006.

Steven Bethard and James H. Martin. CU-TMP: Temporal Relation Classification Using Syntactic and Semantic Features. In *Proceedings of the 4th International Workshop on Semantic Evaluation (SemEval-2007) at ACL 2007*, pages 129–132, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S/S07/S07-1025.

Douglas Biber, Stig Johansson, Geofrey Leech, Susan Conrad, and Edward Finegan. *Longman Grammar of Spoken and Written English*. Longman, 1999.

William Black, Fabio Rinaldi, and David Mowatt. FACILE: Description of the NE system used for MUC-7. In *Proceedings of 7th Message Understanding Conference (MUC-7)*, 1998.

Branimir Boguraev and Rie K. Ando. TimeML-Compliant Analysis of Text Documents. IBM Research Report RC23455, 2004.

Branimir Boguraev and Rie Kubota Ando. TimeBank-Driven TimeML Analysis. In Graham Katz, James Pustejovsky, and Frank Schilder, editors, *Dagstuhl International Workshop on Annotating, Extracting and Reasoning about Time and Events*, number 05151 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2005. Internationales Begegnungs- und Forschungszentrum fur Informatik (IBFI), Schloss Dagstuhl, Germany. URL `http://drops.dagstuhl.de/opus/volltexte/2005/335`.

Branimir Boguraev and Rie Kubota Ando. Analysis of TimeBank as a Resource for TimeML Parsing. In *Proceedings of LREC 2006: Fifth International Conference on Language Resources and Evaluation*, pages 71–76, Genova, Italy, 2006.

Ted Briscoe, Ann Copestake, and Branimir Boguraev. Enjoy the paper: lexical semantics via lexicology. In *Proceedings of 13th International Conference on Computational Linguistics*, pages 42–47, Helsinki, Finland, 1990.

Christopher Burges. A Tutorial on Support Vector Machines for Pattern Recognition. In *Data Mining and Knowledge Discovery*, volume 2, pages 121–167. Kluwer Academic Publishers, 1998.

Stephan Busemann, Thierry Declerck, Abdel Kader Diagne, Luca Dini, Judith Klein, and Sven Schmeier. Natural language dialogue service for appointment scheduling agents. In *Proceedings of the fifth conference on Applied natural language processing*, pages 25–32, Washington, DC, 1997. Association for Computational Linguistics.

Bob Carpenter. Phrasal Queries with LingPipe and Lucene. In *Proceedings of the 13th Meeting of the Text Retrieval Conference (TREC)*, Gaithersburg, Maryland, 2004.

David Cassel, Sarah Taylor, Gary Katz, Lois Childs, and Raymond Rimey. Automated Capture and Representation of Date/Time to Support Intelligence Analysis. In *Intelligence Tools Workshop*, Esbjerg, 2006.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. Classifying Temporal Relations Between Events. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 173–176, Prague, June 2007. Association for Computational Linguistics.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/ cjlin/libsvm/, 2001.

Eugene Charniak, Don Blaheta, Niyu Ge, Keith Hall, John Hale, and Mark Johnson. BLLIP 1987-89 WSJ Corpus Release 1. Linguistic Data Consortium, LDC2000T43, 2000.

Yuchang Cheng, Masayuki Asahara, and Yuji Matsumoto. NAIST.Japan: Temporal Relation Identification Using Dependency Parsed Tree. In *Proceedings of the 4th International Workshop on Semantic Evaluation (SemEval-2007) at ACL 2007*, pages 245–248, Prague, Czech

Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S/S07/S07-1052.

Nancy Chinchor. MUC-7 Named Entity Task Definition (version 3.5). In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.

Timothy Chklovski and Patrick Pantel. VERBOCEAN: Mining the Web for Fine-Grained Semantic Verb Relations. In *Proceedings of EMNLP'2004*, pages 33–40, 2004.

Jacob Cohen. A Coefficient of Agreement for Nominal Scales. In *Educational and Psychological Measurement*, volume 20, pages 37–46. 1960.

Bernard Comrie. *Aspect: An Introduction to the Study of Verbal Aspect and Related Problems*. Cambridge University Press, 1976.

Jim Cowie and Yorick Wilks. *Handbook of Natural Language Processing*, chapter Information Extraction. Marcel Dekker, New York, 2000.

Dan Cristea, Nancy Ide, and Laurent Romary. Veins Theory: a model of global discourse cohesion and coherence. In *Proceedings of the 17th International Conference on Computational Linguistics*, volume 1, pages 281–285. Association for Computational Linguistics, 1998.

Hamish Cunningham, Yorick Wilks, and Robert Gaizauskas. GATE - a General Architecture for Text Engineering. In *Proceedings of the 16th Conference on Computational Linguistics (COLING-96)*, Copenhagen, 1996.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. TiMBL: Tilburg Memory-Based Learner, version 5.1. Reference Guide 04-02, ILK, 2004.

Dagstuhl. Dagstuhl International Seminar on Annotating, Extracting and Reasoning about Times and Events, 2005. URL http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=05151.

David Day, John Aberdeen, Lynette Hirschman, Robyn Kozierok, Patricia Robinson, and Marc Vilain. Mixed Initiative Development of Language Processing Systems. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 348–355, 1997. URL http://www.timeml.org/site/terqas/alembic/AWB-content.html.

David Day, Chad McHenry, Robyn Kozierok, and Laurel Riek. Callisto: A Configurable Annotation Workbench. In *Proceedings of LREC 2004: Fourth International Conference on Language Resources and Evaluation*, 2004. URL http://callisto.mitre.org/.

Leon Derczynski and Robert Gaizauskas. Analysing Temporally Annotated Corpora with CAVaT. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of LREC 2010: Seventh conference on International Language Resources and Evaluation*, Valletta, Malta, 2010a. European Language Resources Association (ELRA).

Leon Derczynski and Robert Gaizauskas. USFD2: Annotating Temporal Expresions and TLINKs for TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation (Semeval-2010) at ACL 2010*, SemEval '10, pages 337–340, Uppsala, Sweden, July 2010b. Association for Computational Linguistics. URL http://aclweb.org/anthology/S/S10/S10-1075.pdf.

David Dowty. *Word Meaning and Montague Grammar: The Semantics of Verbs and Times in Generative Semantics and in Montague's PTQ*. Springer, 1979.

David Dowty. The Effects of Aspectual Class on the Temporal Structure of Discourse: Semantics or Pragmatics. In *Linguistics and Philosophy*, volume 9, pages 37–61. Swets and Zeitlinger, 1986.

Richard Evans. A Comparison of Rule-Based and Machine Learning Methods for Identifying Non-Nominal It. In *Proceedings of Natural Language Processing (NLP 2000)*, pages 233–240, Patras, Greece, 2000. Springer-Verlag.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

Lisa Ferro. The TERN 2004 Evaluation Plan (DRAFT): Time Expression Recognition and Normalization, 2004. URL `http://fofoca.mitre.org/tern_2004/tern_evalplan-2004.29apr04.pdf`.

Lisa Ferro, Beth Sundheim, and George Wilson. TIDES Temporal Annotation Guidelines - Draft Version 1.0. Technical Report Technical Report MTR 00W0000094, MITRE, McLean, Virginia: The MITRE Corporation, 2000.

Lisa Ferro, Inderjeet Mani, Beth Sundheim, and George Wilson. TIDES Temporal Annotation Guidelines. Version 1.0.2. Technical report, MITRE, 2001.

Lisa Ferro, Laurie Gerber, Inderjeet Mani, Beth Sundheim, and George Wilson. TIDES 2003 Standard for the Annotation of Temporal Expressions. Technical report, MITRE, 2003.

Lisa Ferro, Laurie Gerber, Janet Hitzeman, Elizabeth Lima, and Beth Sundheim.

ACE Time Normalization (TERN) 2004 English Training Data v 1.3. Linguistic Data Consortium LDC2004E23, 2004.

Lisa Ferro, Laurie Gerber, Inderjeet Mani, Beth Sundheim, and George Wilson. TIDES 2005 Standard for the Annotation of Temporal Expressions. Technical report, MITRE, April 2005. URL `http://fofoca.mitre.org/annotation_guidelines/timex2_annotation_guidelines.html`.

Elena Filatova and Vasileios Hatzivassiloglou. Domain-Independent Detection, Extraction, and Labeling of Atomic Events. In *Proceedings of RANLP-2003*, 2003.

Elena Filatova and Eduard Hovy. Assigning Time-Stamps to Event-Clauses. In *Proceedings of the ACL 2001 Workshop on Temporal and Spatial Reasoning*, Toulouse, France, 2001.

W. Nelson Francis and Henry Kucera. *Frequency Analysis of English Usage*. Houghton Mifflin, Boston, 1982.

David Graff. The AQUAINT Corpus of English News Text. Linguistic Data Consortium LDC2002T31, 2002. URL `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T31`.

Barbara Grosz and Candace Sidner. Attention, intentions, and the structure of discourse. In *Computational Linguistics*, volume 12, pages 175–204. 1986.

Barbara Grosz, Scott Weinstein, and Aravind Joshi. Centering: A Framework for Modeling the Local Coherence Of Discourse. In *Computational Linguistics*, volume 21, pages 203–225. 1995.

Eun Ha, Alok Baikadi, Carlyle Licata, and James Lester. NCSU: Modeling Temporal Relations with Markov Logic and Lexical Ontology. In *Proceedings of the 5th International Workshop on Semantic Evaluation (Semeval-2010) at ACL 2010*, SemEval '10, pages 341–344, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `http://aclweb.org/anthology/S/S10/S10-1076.pdf`.

Kadri Hacioglu, Ying Chen, and Benjamin Douglas. Automatic Time Expression Labeling for English and Chinese Text. In *Proceedings of CICLing-2005*, volume 3406 of *Lecture Notes in Computer Science*, pages 348–359, Mexico City, Mexico, 2005. Springer-Verlag.

Caroline Hagege and Xavier Tannier. XRCE-T: XIP temporal module for TempEval campaign. In *Proceedings of the 4th International Workshop on Semantic Evaluation (SemEval-2007) at ACL 2007*, pages 492–495, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Sanda Harabagiu and Cosmin Bejan. An Answer Bank for Temporal Inference. In *Proceedings of LREC 2006*, Genoa, Italy, 2006.

Zellig Harris. Distributional Structure. In *Word*, volume 10, pages 146–162. 1954.

Mark Hepple, Andrea Setzer, and Robert Gaizauskas. USFD: Preliminary Exploration of Features and Classifiers for the TempEval-2007 Tasks. In *Proceedings of the 4th International Workshop on Semantic Evaluation (SemEval-2007) at ACL 2007*, pages 438–441, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Janet Hitzeman, Marc Moens, and Claire Grover. Algorithms for Analyzing the Temporal Structure of Discourse. In *Proceedings of the Annual Meeting of the*

European Chapter of the Association of Computational Linguistics (EACL'95), 1995.

Kevin Humphreys, Robert Gaizauskas, Saliha Azzam, Christian Huyck, Brian Mitchell, Hamish Cunningham, and Yorick Wilks. University of Sheffield: Description of the University of Sheffield LaSIE-II System as used for MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Morgan Kaufmann, 1999.

Ben Hutchinson. Acquiring the meaning of discourse markers. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, ACL '04, pages 685–692, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

ISO-TimeML. Language resource management - Semantic annotation framework (SemAF) - Part 1: Time and events, 2007. URL `http://lirics.loria.fr/doc_pub/SemAFCD24617-1Rev12.pdf`.

ISO8601:2004. Data elements and interchange formats - Information interchange - Representation of dates and times, 2004.

Abraham Ittycheriah, Lucian Lita, Nanda Kambhatla, Nicolas Nicolov, Salim Roukos, and Margo Stys. Identifying and Tracking Entity Mentions in a Maximum Entropy Framework. In *Proceedings of HLT-NAACL 2003*, Edmonton, Canada, 2003.

Ray Jackendoff. *Semantic Structures*. The MIT Press, 1990.

Hubert Jin, Rich Schwartz, Sreenivasa Sista, and Frederick Walls. Topic Tracking

for Radio, TV Broadcast, and Newswire. In *Proceedings of the DARPA Broadcast News Workshop*, Herndon, Virginia, 1999.

Immanuel Kant. *Critique of pure reason*. Hackett Publishing Co, 1999.

Graham Katz and Fabrizio Arosio. The Annotation of Temporal Information in Natural Language Sentences. In *Proceedings of the ACL 2001 Workshop on Temporal and Spatial Information Processing*, pages 104–111, Toulouse, France, 2001. Association for Computational Linguistics.

Judith Klavans and Martin Chodorow. Degrees of Stativity: The Lexical Representation of Verb Aspect. In *Proceedings of COLING-1992*, pages 1126–1131, 1992.

Anup-Kumar Kolya, Asif Ekbal, and Sivaji Bandyopadhyay. JU_CSE_TEMP: A First Step towards Evaluating Events, Time Expressions and Temporal Relations. In *Proceedings of the 5th International Workshop on Semantic Evaluation (Semeval-2010) at ACL 2010*, SemEval '10, pages 345–350, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `http://aclweb.org/anthology/S/S10/S10-1077.pdf`.

John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the International Conference on Machine Learning*, 2001.

Mirella Lapata and Alex Lascarides. Inferring Sentence-Internal Temporal Relations. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 153–160, 2004.

Mirella Lapata and Alex Lascarides. Learning Sentence-internal Temporal

Relations. In *Journal of Artificial Intelligence Research*, number 27, pages 85–117. 2006.

Chong Min Lee and Graham Katz. Error Analysis of the TempEval Temporal Relation Identification Task. In *Proceedings of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 138–145, Boulder, Colorado, June 2009. Association for Computational Linguistics.

Wenjie Li, Kam-Fai Wong, Guihong Cao, and Chunfa Yuan. Applying Machine Learning to Chinese Temporal Relation Resolution. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 582–588, 2004.

Hector Llorens, Estela Saquete, and Borja Navarro. TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation (Semeval-2010) at ACL 2010*, SemEval '10, pages 284–291, Uppsala, Sweden, July 2010a. Association for Computational Linguistics. URL `http://aclweb.org/anthology/S/S10/S10-1063.pdf`.

Hector Llorens, Estela Saquete, and Borja Navarro-Colorado. TimeML Events Recognition and Classification: Learning CRF Models with Semantic Roles. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 725–733, Beijing, August 2010b.

Robert Longacre. *The Grammar of Discourse*. New York: Plenum Press, 1983.

LREC-2002. Workshop on Annotation Standards for Temporal Information in Natural Language, 2002. URL `http://www.lrec-conf.org/lrec2002/lrec/wksh/Annotation.html`.

Inderjeet Mani. Recent developments in temporal information extraction. In *Proceedings of RANLP 2003 on Recent Advances in Natural Language Processing*, pages 45–60, 2003.

Inderjeet Mani and Barry Shiffman. Temporally Anchoring and Ordering Events in News. In James Pustejovsky and Robert Gaizauskas, editors, *Time and Event Recognition in Natural Language*. John Benjamins, Amsterdam, 2005.

Inderjeet Mani and George Wilson. Robust Temporal Processing of News. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 69–76, Hong Kong, 2000.

Inderjeet Mani, Barry Schiffman, and Jianping Zhang. Inferring Temporal Ordering of Events in News. In *Proceedings of the Human Language Technology Conference (HLT-NAACL'03)*, 2003.

Inderjeet Mani, James Pustejovsky, and Beth Sundheim. Introduction to the Special Issue on Temporal Information Processing. In *ACM Transactions on Asian Language Processing: Special issue on Temporal Information Processing*, volume 3, pages 1–10, 2004.

Inderjeet Mani, James Pustejovsky, and Robert Gaizauskas, editors. *The Language of Time: A Reader*. Oxford University Press, 2005.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. Machine Learning of Temporal Relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 753–760, Sydney, Australia, July 2006. Association for Computational Linguistics.

Inderjeet Mani, Ben Wellner, Marc Verhagen, and James Pustejovsky. Three Approaches to Learning TLINKs in TimeML. Technical Report CS-07-268, The MITRE Corporation, Computer Science Department, Brandeis University, 2007.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. URL `http://nlp.stanford.edu/IR-book/`.

Olivia March and Timothy Baldwin. Automatic event reference identification. In *Proceedings of the 2008 Australasian Language Technology Workshop (ALTW 2008)*, pages 79–87, Hobart, Australia, 2008.

Pawel Mazur and Robert Dale. A Rule Based Approach to Temporal Expression Tagging. In *Proceedings of the International Multiconference on Computer Science and Information Technology*, pages 293–303, 2007.

John McTaggart. The Unreality of TIme. In *Mind*, volume 17, pages 457–473. Oxford University Press, 1908.

Andrei Mikheev, Claire Grover, and Marc Moens. Description of the LTG system used for MUC-7. In *Proceedings of 7th Message Understanding Conference (MUC-7)*, 1998.

Congmin Min, Munirathnam Srikanth, and Abraham Fowler. LCC-TE: A Hybrid Approach to Temporal Relation Identification in News Text. In *Proceedings of the 4th International Workshop on Semantic Evaluation (SemEval-2007) at ACL 2007*, pages 219–222, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Ruslan Mitkov. *The Oxford Handbook of Computational Linguistics*, chapter Anaphora Resolution, pages 269–275. Oxford University Press, 2003.

Marc Moens and Mark Steedman. Temporal Ontology and Temporal Reference. In *Computational Linguistics*, volume 14, pages 15–28. June 1988.

Dan Moldovan, Christine Clark, and Sanda Harabagiu. Temporal Context Representation and Reasoning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-2005)*, pages 1099–1104, 2005.

Matteo Negri and Luca Marseglia. Recognition and Normalization of Time Expressions: ITC-irst at TERN 2004. Technical Report Technical Report WP3.7, ITC-irst, February 2005.

Constantin Orăsan. *Comparative Evaluation of Modular Automatic Summarisation Systems Using CAST*. PhD thesis, University of Wolverhampton, 2006.

Terence Parsons. *Events in the Semantics of English*. The MIT Press, Cambridge, 1990.

Jordi Poveda, Mihai Surdeanu, and Jordi Turmo. An Analysis of Bootstrapping for the Recognition of Temporal Expressions. In *Proceedings of the NAACL HLT Workshop on Semi-supervised Learning for Natural Language Processing*, pages 49–57, Boulder, Colorado, 2009. Association for Computational Linguistics.

John Prager, Eric Brown, and Anni Coden. Question-Answering by Predictive Annotation. In *Proceedings of the 23rd Annual International ACM SIGIR*

Conference on Research and Development in Information Retrieval, pages 184–191, Athens, Greece, July 2000.

Georgiana Puşcaşu. A Multilingual Method for Clause Splitting. In *Proceedings of the 7th Annual Colloquium for the UK Special Interest Group for Computational Linguistics (CLUK 2004)*, pages 199–206, Birmingham, United Kingdom, 2004a.

Georgiana Puşcaşu. A Framework for Temporal Resolution. In *Proceedings of the 4th Conference on Language Resources and Evaluation (LREC 2004)*, pages 1901–1904, Lisbon, Portugal, 2004b. URL `http://clg.wlv.ac.uk/papers/puscasu_lrec04.pdf`.

Georgiana Puşcaşu. Discovering Temporal Relations with TicTac. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2007)*, pages 493–498, Borovets, Bulgaria, 2007a.

Georgiana Puşcaşu. WVALI: Temporal Relation Identification by Syntactico-Semantic Analysis. In *Proceedings of the 4th International Workshop on Semantic Evaluation (SemEval-2007) at ACL 2007*, pages 484–487, Prague, Czech Republic, June 2007b. Association for Computational Linguistics. URL `http://aclweb.org/anthology/S/S07/S07-1108.pdf`.

Georgiana Puşcaşu and Verginica Barbu-Mititelu. Annotation of WordNet Verbs with TimeML Event Classes. In *Proceedings of the 6th Conference on Language Resources and Evaluation (LREC 2008)*, pages 2793–2800, Marrakech, Morocco, 2008.

Georgiana Puşcaşu and Ruslan Mitkov. If it were then, then when was it? Establishing the anaphoric role of then. In *Proceedings of the 5th Conference on*

Language Resources and Evaluation (LREC 2006), pages 1194–1199, Genoa, Italy, 2006.

Georgiana Puşcaşu, Patricio Martinez-Barco, and Estela Saquete-Boro. On the Identification of Temporal Clauses. In *Proceedings of the 5th Mexican International Conference on Artificial Intelligence (MICAI 2006)*, pages 911–921, Apizaco, Mexico, 2006.

James Pustejovsky. *The Generative Lexicon*. The MIT Press, 1995.

James Pustejovsky, José Casta no, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Proceedings of IWCS-5: Fifth International Workshop on Computational Semantics*, 2003.

James Pustejovsky, Marc Verhagen, Roser Saurí, Jessica Littman, Robert Gaizauskas, Graham Katz, Inderjeet Mani, Robert Knippen, and Andrea Setzer. TimeBank 1.2. Linguistic Data Consortium LDC2006T08, 2006. URL `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T08`.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. *A Comprehensive Grammar of the English Language*. Longman, 1985.

Malka Rappaport Hovav and Beth Levin. *The Projection of Arguments: Lexical and Compositional Factors*, chapter Building Verb Meanings, pages 97–134. CSLI Publications, Stanford, CA, 1998.

Adwait Ratnaparkhi. Learning to Parse Natural Language with Maximum Entropy Models. In *Machine Learning*, volume 34, pages 151–175. Kluwer Academic Publishers, 1999.

Hans Reichenbach. *Elements of Symbolic Logic*. The Macmillan Company, New York, 1947.

Tony G. Rose, Mark Stevenson, and Miles Whitehead. The Reuters Corpus Volume 1 - from Yesterday's News to Tomorrow's Language Resources. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*, pages 827 – 833, Las Palmas de Gran Canaria, May 2002.

Susan Rothstein. *Structuring Events: A Study in the Semantics of Lexical Aspect*. Wiley-Blackwell, 2004.

Robert Rynasiewicz. *The Stanford Encyclopedia of Philosophy*, chapter Newton's Views on Space, Time, and Motion. Stanford University, 2004. URL `http://plato.stanford.edu/`.

Geoffrey Sampson. *English for the computer: the SUSANNE corpus and analytic scheme*. Oxford University Press, 1995.

Estela Saquete-Boro. *Temporal Expression Recognition and Resolution applied to Event Ordering*. PhD thesis, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, 2005.

Roser Saurí. *A Factuality Profiler for Eventualities in Text*. PhD thesis, Brandeis University, 2008.

Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. Evita: A Robust Event Recognizer for QA Systems. In *Proceedings of HLT/EMNLP 2005*, pages 700–707, 2005.

Roser Saurí, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. TimeML Annotation Guidelines Version 1.2.1, 2006. URL `http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf`.

Deborah Schiffrin. Between text and context: Deixis, anaphora and the meaning of then. In *Text - Interdisciplinary Journal for the Study of Discourse*, volume 10, pages 245–270. Walter de Gruyter, 1990.

Frank Schilder and Christopher Habel. From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages. In *Proceedings of the ACL 2001 Workshop on Temporal and Spatial Information Processing*, pages 65–72, Toulouse, France, 2001. Association for Computational Linguistics.

Andrea Setzer. *Temporal Information in Newswire Articles: An Annotation Scheme and Corpus Study*. PhD thesis, University of Sheffield, 2001.

Andrea Setzer and Robert Gaizauskas. On the Importance of Annotating Temporal Event-Event Relations in Text . In *Proceedings of the LREC 2002 Workshop on Annotation Standards for Temporal Information in Natural Language*, pages 52–60, 2002.

Eric Siegel and Kathleen McKeown. Learning Methods to Combine Linguistic Indicators: Improving Aspectual Classification and Revealing Linguistic Insights. In *Computational Linguistics*, volume 26, pages 595–627. Association for Computational Linguistics, 2001.

Carlota Smith. *The Parameter of Aspect*. Kluwer Academic Press, 1991.

Catherine Soanes and Angus Stevenson, editors. *Oxford Dictionary of English*. Oxford University Press, 2005.

Beth Sundheim. Tipster/MUC-5: information extraction system evaluation. In *MUC5 '93: Proceedings of the 5th conference on Message understanding*, pages 27–44, Baltimore, Maryland, 1993. Association for Computational Linguistics.

Beth Sundheim and Nancy Chinchor. Survey of the Message Understanding Conferences. In *Proceedings of the Workshop on Human Language Technology*. Association for Computational Linguistics, 1993.

Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. Using Predicate-Argument Structures for Information Extraction. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 8–15, July 2003.

TANGO. TimeML Annotation Graphical Organizer, ARDA Workshop on Advanced Question Answering Technology, 2003. URL `http://www.timeml.org/site/tango/index.html`.

Pasi Tapanainen and Timo Jarvinen. A Non-Projective Dependency Parser. In *Proceedings of the 5th Conference of Applied Natural Language Processing*, pages 64–71, Washington DC, USA, March 1997.

Carol Tenny. *Grammaticalizing aspect and affectedness*. PhD thesis, MIT, 1987.

Carol Tenny and James Pustejovsky. *Events as Grammatical Objects*, chapter A History of Events in Linguistic Theory, pages 3–37. CSLI Publications, 2000.

Alice ter Meulen. *Representing Time in Natural Language: The Dynamic Interpretation of Tense and Aspect*. The MIT Press, Cambridge, MA, 1995.

TERQAS. Time and Event Recognition for Question Answering Systems, ARDA Workshop on Advanced Question Answering Technology, 2002. URL `http://www.timeml.org/site/terqas/index.html`.

Ellen Thompson. *Time in Natural Language: Syntactic Interfaces with Semantics and Discourse*. Walter de Gruyter, 2005.

TIDES. DARPA Program in Translingual Information Detection Extraction and Summarization, 2002. URL `http://infowar.net/tia/www.darpa.mil/iao/TIDES.htm`.

TIME-2006. International Symposium on Temporal Representation and Reasoning, 2006. URL `http://www.timeml.org/time2006/`.

Naushad UzZaman and James Allen. TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text. In *Proceedings of the 5th International Workshop on Semantic Evaluation (Semeval-2010) at ACL 2010*, SemEval '10, pages 276–283, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `http://aclweb.org/anthology/S/S10/S10-1062.pdf`.

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Information Science and Statistics. Springer-Verlag, New York, USA, 1995.

Argyrios Vasilakopoulos and William J. Black. Temporally ordering event instances in natural language texts. In *Proceedings of the International*

Conference on Recent Advances in Natural Language Processing (RANLP 2005), 2005.

Zeno Vendler. *Linguistics in Philosophy*, chapter Verbs and times, pages 97–121. Cornell University Press, 1967.

Marc Verhagen. *Times Between The Lines*. Ph.d. dissertation, Department of Computer Science, Brandeis University, Waltham, MA, USA, 2004.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. SemEval-2007 Task 15: TempEval Temporal Relation Identification. In *Proceedings of the 4th International Workshop on Semantic Evaluation (Semeval-2007) at ACL 2007*, pages 75–80, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://aclweb.org/anthology/S/S07/S07-1014.pdf`.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Jessica Moszkowicz, and James Pustejovsky. The TempEval Challenge: Identifying Temporal Relations in Text. In *Language Resources and Evaluation*, volume 43, pages 161–179. 2009.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation (Semeval-2010) at ACL 2010*, SemEval '10, pages 57–62, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `http://aclweb.org/anthology/S/S10/S10-1010.pdf`.

Henk Verkuyl. A Theory of Aspectuality. The Interaction between Temporal and Atemporal Structure. In *Cambridge Studies in Linguistics*, volume 64. Cambridge University Press, 1993.

Bonnie Webber. Tense as Discourse Anaphor. In *Computational Linguistics*, volume 14, pages 61–73. Association for Computational Linguistics, 1988.

Janyce Wiebe, Thomas O'Hara, Thorsten Ohrstrom-Sandgren, and Kenneth McKeever. An Empirical Approach to Temporal Reference Resolution. In *Journal of Artificial Intelligence Research*, number 9, pages 247–293. 1998.

Roman Yangarber and Ralph Grishman. NYU: Description of the Proteus/PET System as Used for MUC-7 ST. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1999.

Tong Zhang, Fred Damerau, and David Johnson. Text Chunking based on a Generalization of Winnow. In *Journal of Machine Learning Research*, volume 2, pages 615–637. 2002.